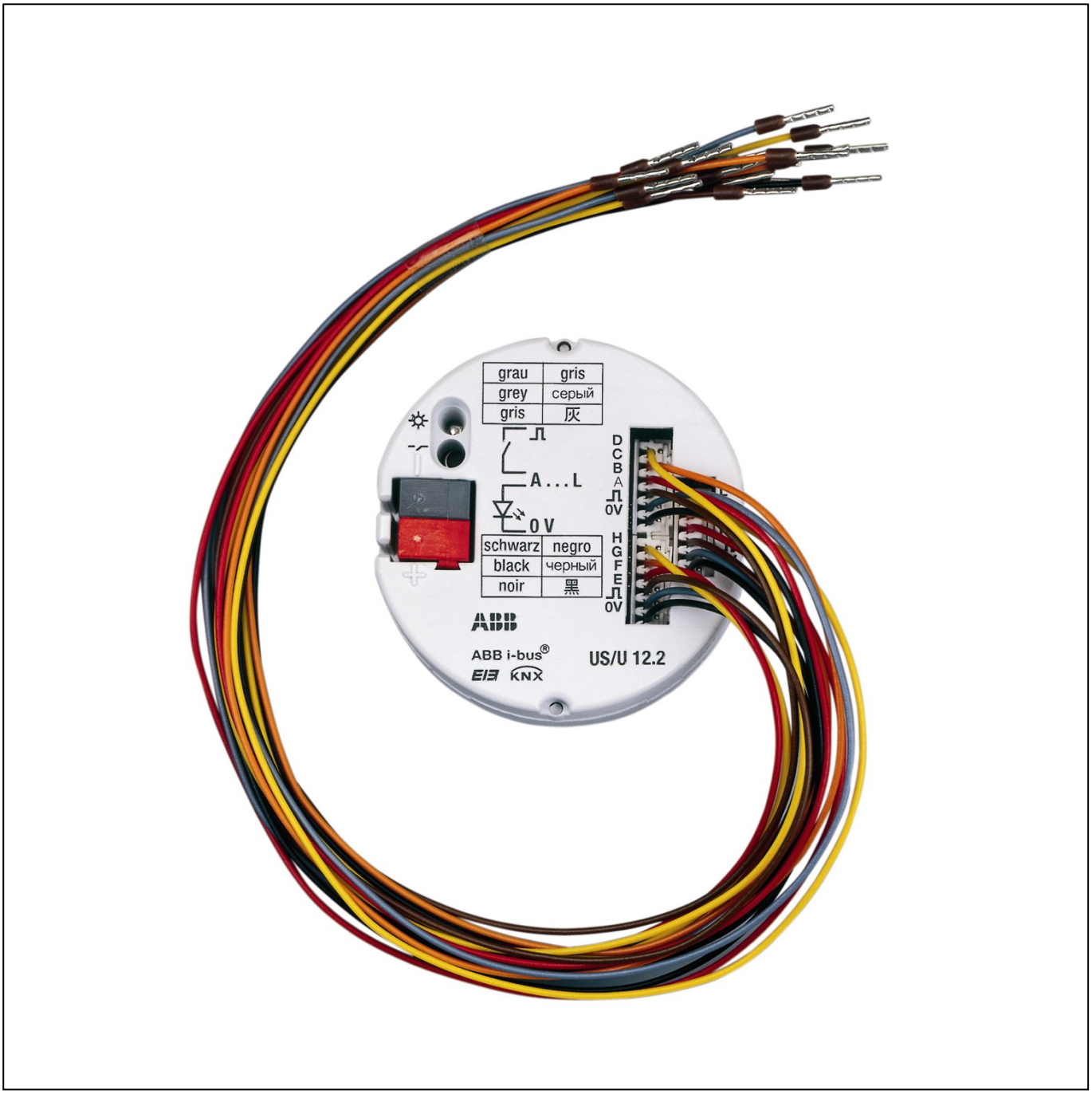
Product Manual

ABB i-bus® KNX

Universal Interface US/U 12.2

Intelligent Installation Systems



Contents

Page

This manual describes the function of the Universal Interface US/U 12.2 with its application program *Binary Input Display Heat 12f/1*.
Subject to changes and errors excepted.

Exclusion of liability:
Despite checking that the contents of this document match the hardware and software, deviations cannot be completely excluded. We therefore cannot accept any liability for this. Any necessary corrections will be incorporated in new versions of the manual.
Please inform us of any suggested improvements.

1 General

The Universal Interface US/U 12.2 is used for the operation and display of building functions via push buttons and light emitting diodes. The compact design facilitates the device to be inserted in a conventional 60 mm wiring box, e.g. behind an operating panel.

This manual provides you with detailed technical information relating to the device, its installation and programming. Furthermore, you will find application examples for effective device usage in the last section of the manual.

1.1 Product and functional overview

The Universal Interface US/U 12.2 has twelve channels for connection of conventional push buttons (input operation) or LEDs (output operation). Alternatively, an Electronic Relay ER/U 1.1 can be connected to each channel.

The operating mode of every channel can be parameterized individually. Every connecting cable is approx. 30 cm in length and can be extended up to max. 10 m.

The power supply for the LEDs (2 mA per channel) is provided by the device. No additional power supply is required.

General

An exceptionally comprehensive and clearly arranged functionality permits usage in the most differing fields of application. The following list provides an overview:

- Switching and dimming of lighting (also for 1-button operation)
- Operation of blinds and roller blinds (also for 1-button operation)
- Sending of arbitrary values, e.g. temperature values
- Control and storing of light scenes
- Triggering an electronic relay for control of electro-thermal valve drive for heating valves
- Control of an LED (with flashing function and time limitation) for feedback of an operation
- Operation of different loads by multiple push button actions
- Operation of several loads in a fixed switching sequence
- Reading out of technical contacts (e.g. relays)

Each channel of a device can assume one of the functions described above.

2 Device technology

The device has 12 channels, which can be parameterized individually in the ETS, as inputs or outputs.

Using the colour-coded connecting cables, it is possible to connect conventional push buttons, floating contacts or LEDs.

The contact scanning voltage and the supply voltage for the LED's are made available by the device. Series resistors for operation of the LEDs are also integrated into the device.

The Universal Interface is installed in a conventional 60 mm combined wall and joint box.

The bus connection is established using the enclosed bus connecting terminal.

2.1 Technical data

Power supply:	- Rated voltage	21...30 V DC, via the bus
	- Current consumption	10 mA
Inputs/outputs	- Number	12, can be separately parametrized as inputs or outputs
	- Permitted cable length	≤ 10 m
Input:	- Sensing voltage U_n	20 V DC (pulsed)
	- Input current I_n	0.5 mA
Output:	- Output voltage	3.3 V DC
	- Output current	Max. 2 mA
	- Safety	Short-circuit protected, overload protected, reverse voltage polarity protected
Operating and display elements	- LED (red) and button	For assignment of the physical address
Connections	- Inputs/outputs	3x6 cables approx. 30 cm long, can be extended to max. 10 m
	- KNX	Bus connection terminal
Temperature range	- Operation	-5° C ... + 45° C
	- Storage	-25° C ... + 55° C
	- Transport	-25° C ... + 70° C
Enclosure	IP 20 in the installed state	Compliant to EN 60 529
Safety class	III	Compliant to EN 61 140
Installation	In switch box & 60mm	
Mounting position	As required	
Dimensions (B x H)	54 x 19 mm	
Weight	0.06 kg	
Enclosure	Plastic, halogen free, colour grey	
Approval	KNX to EN 50 090-1, -2	
CE mark	In accordance with the EMC guideline and low voltage guideline	

Application program	Number of communication objects	Max. number of group addresses	Max. number of associations
Binary Input Display Heat 12f/1	84	254	255

Note

The ETS and the current version of the device application program are required for programming.

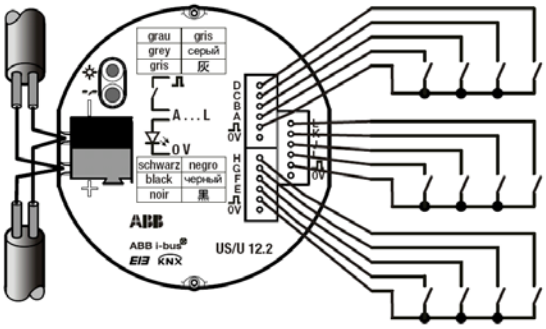
The current application program can be found with the respective software information for download on the Internet at *www.abb.com/knx*. After import in the ETS, it is available in the ETS under *ABB/Folder*.

The device does not support the locking function of a KNX device in the ETS. If you inhibit access to all devices of the project with a *BCU code*, it has no effect on this device. Data can still be read and programmed.

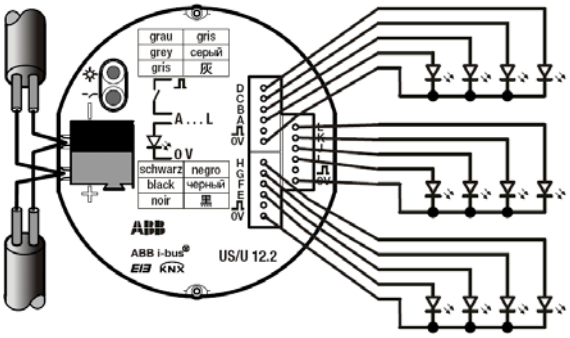
2.2 Circuit diagram

The maximum cable length is 10 m. The colours of the connection cables are explained in section 2.5.

Connection of a floating push button/switch:



Connection of light emitting diodes (LEDs)



Series resistors for operation of the LEDs are integrated into the device. The maximum output current is 2 mA per LED.

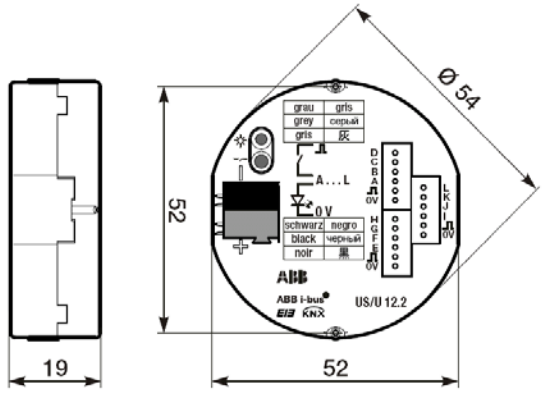
Connection of an Electronic Relay ER/U 1.1

An Electronic Relay is connected accordingly to an LED: The coloured core is connected to “+”, the black core is connected to “–”.

Important: Other relay types other than the ER/U 1.1 cannot be controlled!

Note: The connection to an S0 pulse output is only possible for ABB electronic energy meters. The correct polarity must be observed (“+” to grey core, “–” to coloured core).

2.3 Dimension drawing



2.4 Assembly and installation

The mounting position of the device can be selected as required. Any cores not required must be insulated.

Accessibility to the device for the purpose of operation, testing, visual inspection, maintenance and repair must be must be provided (compliant to VDE 0100-520).

2.5 Description of the inputs and outputs

Grey core (⎍): Positive scanning voltage

When operated as an input, the grey core provides the positive, pulsed scanning voltage.

Coloured core: Control of the channel

When operated as an input, the state of the contact is read out via the coloured cores.

When operated as an output, the coloured core provides the positive output voltage.

The following table allocates the colours to the channels:

brown	Channel A, E and I
red	Channel B, F and J
orange	Channel C, G and K
yellow	Channel D, H and L

Black core (0 V): Negative reference potential

When operated as an output, the black core provides the negative reference potential.

Important: The inputs and outputs are not electrically isolated from the KNX bus voltage (SELV). The SELV criteria only allow the connection of floating contacts, which feature a safe electrical isolation.

3 Commissioning

3.1 Overview

The Universal Interface US/U 12.2 features a high-performance application program "Binary Input Display Heat *12f/1"*. The following operating modes can be set separately for each input:

Switch sensor	For switching the lighting or reading a floating contact (relay) Distinction between short/long operation and cyclical sending of the contact state are possible.
Switch/dimming sensor	For switching/dimming the lighting Start-stop dimming and stepwise dimming as well as dimming via a single push button are possible.
Shutter sensor	For movement/louvre adjustment of a blind or a shutter Eight preset operating responses are possible in total.
Value / forced operation	For sending values of different data types (e.g. temperature values) It is possible to send different values or data types after a short/long operation, activation/deactivation of the forced operation of actuators is also possible
Control scene	For recalling and storing the states of several actuator groups The actuator groups can either be controlled via max. 5 individual objects or (if supported by the actuators) via a special 8 bit scene object.
Control electr. relay (heating actuator)	For control of an electro-thermal valve drive. The device is controlled via an Electronic Relay ER/U 1.1, which is connected between the US/U and the electro-thermal valve drive. The device has the full functionality of a heating actuator. Control via 2-step (ON-OFF) controller or continuous controller (PWM), cyclical valve purging, monitoring of the room thermostat and forced operation of the valve drive are possible.

Control LED	For controlling an LED Switching and flashing (time limited and at different flashing frequencies) as well as usage as an orientation light are possible.
Switching sequence	For the operation of several actuator groups consecutively. The actuators are switched in preset sequences. A selection between different sequences can be made.
Multiple operation	For triggering of different functions depending on the frequency of actuation For example, all the illumination in a room can be switched on by pressing the button twice, whereas pressing the button once will only switch individual lamps. Even a long actuation can be detected.

Supplied state

The device is assigned with the physical address 15.15.255 in the factory. The application program is preloaded in the factory. If required, the application program can be reloaded by discharging the device entirely.

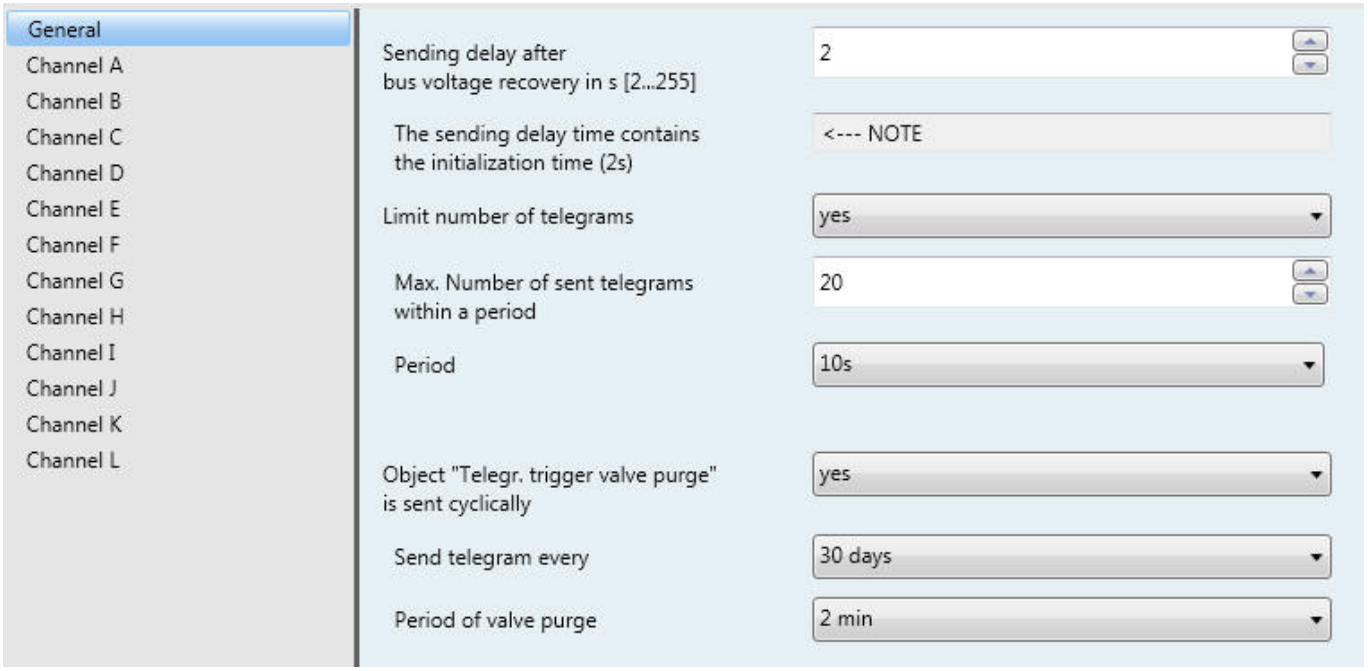
A longer downtime may result if the application program is changed or after a discharge.

3.2 Parameters and communication objects

3.2.1 General parameters

Parameters for the functions, which concern the complete device can be set via the “General” parameter window.

3.2.1.1 Parameter window “General”



Sending delay after after bus voltage recovery in s [2...255s]	2…255
Limit number of telegrams	yes no
Max. Number of sent telegrams within a period	0…20…255
Period	50 ms…500 ms…1 s…10 s/30 s / 1 min
Object "Telegr. trigger valve purge" is sent cyclically	yes no
Send telegram every	7 / 14 / 30 / 50 days
Period of valve purge	1 / 2 / 5 / 10 min

Sending delay
after bus voltage recovery in s [2...255s]

The transmision delay time determines the period between bus voltage recovery and the point after which telegrams can be sent. An initialisation period of approx. 2 seconds for starting the device is included in the delay time.

If object value are read out via the bus during the sending delay (e.g. from visualisation terminals), these requests are stored and are answered once the sending delay has elapsed.

How does the device behave after bus voltage recovery?

After bus voltage recovery, the object values are generally reset to value “0”. Depending on the operating mode, the object values can then be updated (see below).

There is a sending delay before telegrams are sent on the bus. The following diagram indicates the time sequence:

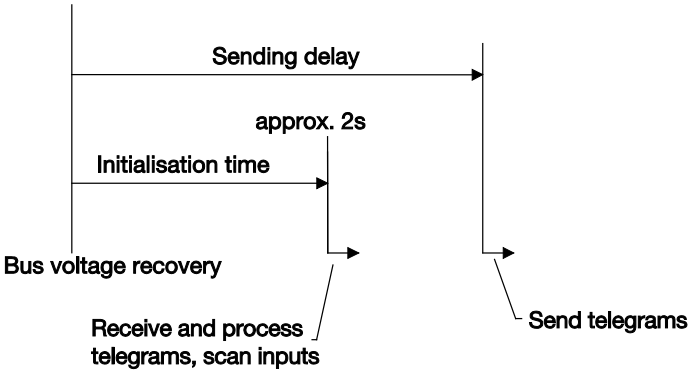


Fig. 1: Reaction after bus voltage recovery

The inputs are scanned after the initialisation period and the object values are updated if the parameterization has been set accordingly. If the input is actuated at bus voltage recovery, the device will behave as if the actuation commenced at the end of the initialisation time.

Particular behaviour of the individual operating modes

Operating mode	**Reaction after bus voltage recovery**
Switch sensor	The reaction can be set in the parameters. However, if there is a distinction between a short and a long operation or the value “TOGGLE” has been set in one of the parameters “Reaction on closing/opening the contact”, no telegrams are sent after bus voltage recovery.
Switch/dimming sensor	If the input is actuated at bus voltage recovery, the device ends the corresponding telegram on the bus. Otherwise no telegram is sent.
Shutter sensor	If the input is actuated at bus voltage recovery, the device ends the corresponding telegram on the bus. Otherwise no telegram is sent.
Value / forced operation	The reaction can be set in the parameters. The object values are overwritten (updated) by the parameterized values at bus voltage recovery. If an object value has been changed via the bus beforehand, this value will be lost.
Control scene	When the scene is controlled via “5 separate objects”, the object values of the scene are overwritten with the parameterized values.
Control electr. relay (heating actuator)	Until the first telegram of the room thermostat has been received, the parameterised value is set.
Control LED	The output state can be set in the parameters.
Switching sequence	If the input is actuated at bus voltage recovery, the device ends the corresponding telegram on the bus. Otherwise no telegram is sent.
Multiple operation	If the input is actuated at bus voltage recovery, the device ends the corresponding telegram on the bus. Otherwise no telegram is sent.

Limit number of telegrams

In order to check the bus load, which is generated by the device, there is a powerful limit function for telegrams. It is possible to set how many telegrams ("***Max. number of telegrams within a period***") can be sent within an adjustable period ("***Period***").

How does the telegram rate limitation function?

A new monitoring period starts at the end of the previous monitoring period. The sent telegrams are counted. As soon as the Max. number of sent telegrams within a period has been reached, no further telegrams are sent on the bus until the end of the monitoring period. When a new monitoring period starts, the telegram counter is reset to zero and the sending of telegrams is again permitted.

Object "Telegr. trigger valve purge" is sent cyclically

This function is only relevant if device is used to control an electronic relay. Regular purging of a heating valve can prevent deposits from forming in the valve area and restricting the valve function. This is particularly important at times when the valve position does not change very much.

If this parameter is set to yes, the object "Telegr. trigger valve purge" is visible, so that it is sent at adjustable intervals to start the valve purge ("***repeat valve purge every***") and has the value "1" for the "***Period of valve purge***". The object "Valve purge" of a channel, which has been parameterized with the function of a heating actuator, can be controlled via this object.

3.2.1.2 General communication objects

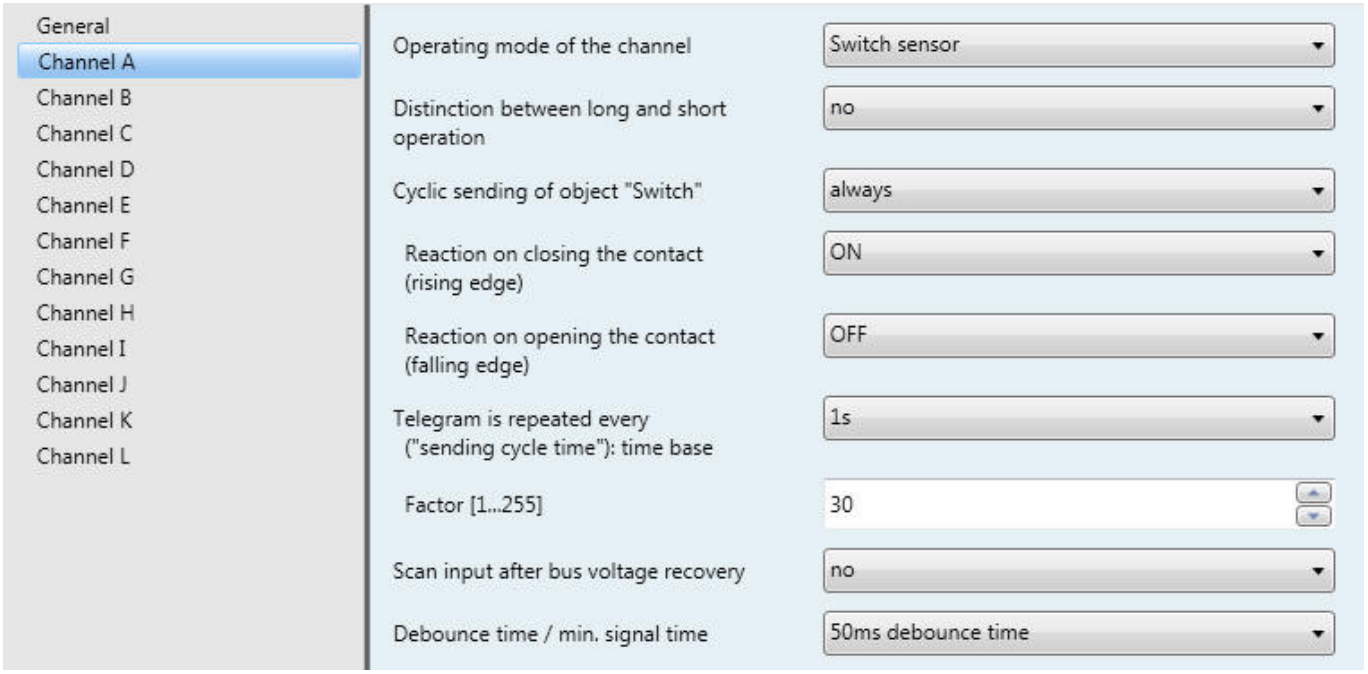
No.	Function	Object name	Data type	Flags
85	**Telegr. trigger valve purge**	**General**	**1 bit** **DPT 1.001**	**C, T**
The object is set at regular intervals ("*Period of valve purge*") to the value "1" for an adjustable period and then reset to "0". It can be used, for example, to trigger a valve purge at regular intervals (see object "Valve purge"). After bus voltage recovery, this object sends the value "0" on the bus and the purge cycle is restarted. This object is visible if the parameter "*Transmit object "Telegr. valve purge*" is set to yes.				

3.2.2 Operating mode “Switch sensor”

The “Switch sensor” mode is described in the following.

3.2.2.1 Parameter “without short/long operation”

The following parameters are visible, if the parameter *distinction between long and short operation* has been set to the value *no*:



Operating mode of the channel	Switch sensor
Distinction between long and short operation	yes no
Cyclic sending of object “Switch”	no if "Switch" = ON if "Switch" = OFF always
Reaction on closing the contact (rising edge)	ON OFF TOGGLE no reaction
Reaction on opening the contact (falling edge)	ON OFF TOGGLE no reaction

Telegram is repeated every ("sending cycle time"): time base	1 s / 10 s / 1 min / 10 min / 1h
Factor [1…255]	1…30…255
Scan input after bus voltage recovery	yes no
Debounce time / min. signal time	10 ms…50 ms…150 ms debounce time Minimum signal time

Distinction between long and short operation

If the parameter is set to *no,* the input will be evaluated normally on every edge of the input signal. Evaluation is undertaken immediately.

If *yes* is selected, there is a delay after opening/closing the contact to determine whether there is a short or long operation. Only thereafter will a possible reaction be triggered.

The following table shows the function in detail:

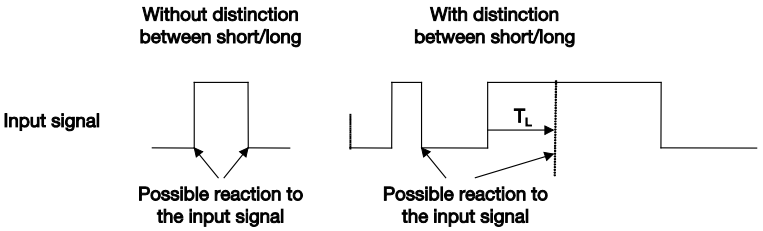


Fig. 2: Distinction between short/long operation for “Switch sensor” mode

T_L is the time duration from where a long operation is detected.

Cyclic sending of object “Switch”

This parameter is visible if there is no distinction between short and long actuation.

Option *always*: The object sends cyclically on the bus, regardless of its value.

Option *if “Switch” = ON*: Only the value “1” is sent cyclically

Option *if “Switch” = OFF*: Only the value “0” is sent cyclically

How does cyclic sending function?

Cyclic sending enables the communication object *Switch* to send automatically on the bus at a fixed interval.

If cyclic sending is only carried out for a specific object value (ON or OFF), this condition refers to the value of the communication object. It is therefore possible in principle to start cyclic sending by sending a value to the communication object *Switch*. As this reaction is generally unwanted, the “write” flag and “update” flag of the communication object have to be deleted as standard in the setting to ensure that it cannot by changed via the bus. If this functionality is still required however, these flags should be set accordingly.

When the *Switch* object changes and after bus voltage recovery, the object value is sent immediately on the bus and the sending cycle time restarts.

Why is this function required?

This function can be used, for example, to monitor life signs from the sensor.

Reaction on closing the contact
Reaction on opening the contact

This parameter is visible if there is no distinction between short and long actuation. 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 parameterised, it is still possible to select the option “terminate cyclic sending”. It is thus possible to stop cyclic sending without a new object value being sent.

Telegram is repeated every ("sending cycle time")

This parameter is visible if cyclical sending has been set. It describes the interval between two telegrams that are sent cyclically:

Sending cycle time = Time base x Factor

Scan input after bus voltage recovery

It can be set whether the current status of the input is sent on the bus (via object *Switch*) after bus voltage recovery (once the sending delay has elapsed).

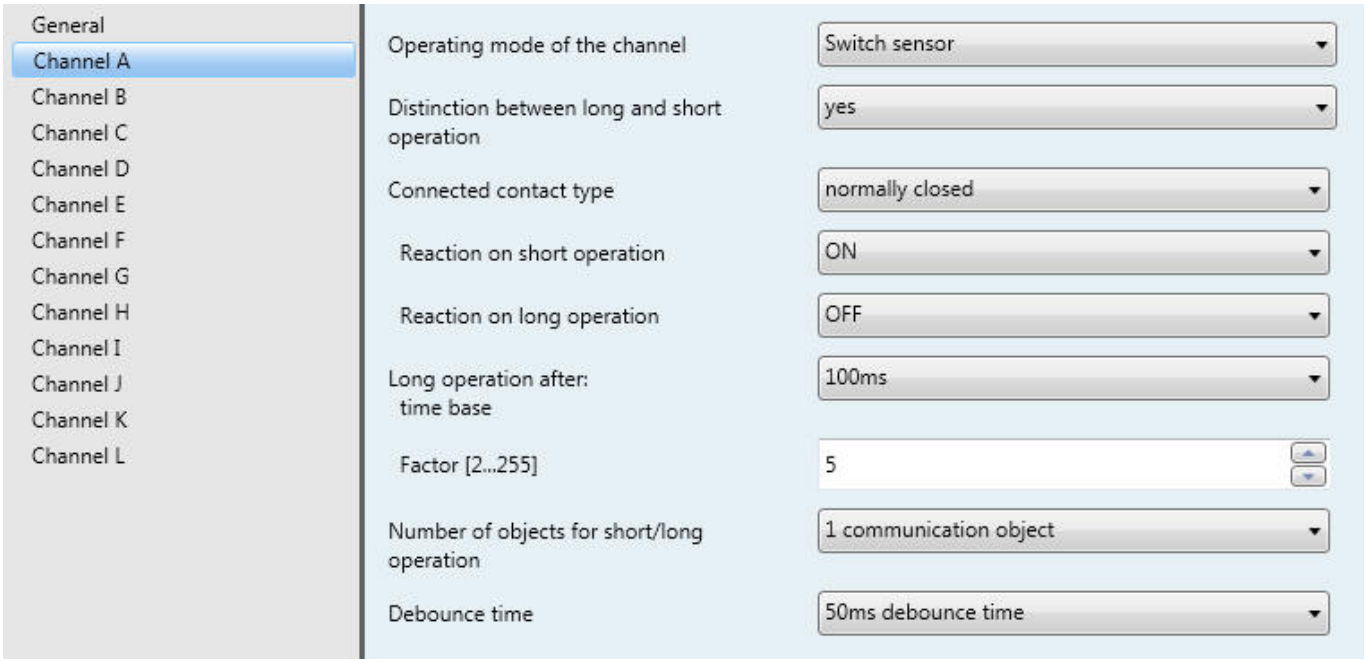
A value is however, only sent on the bus if the value *TOGGLE* has not been set in either of the two parameters *Reaction on opening/closing the contact.* If one of the two parameters has the value *TOGGLE*, no values are sent in general on the bus after bus voltage recovery.

Debounce time / min. signal time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

3.2.2.2 Parameters “with short/long operation”

The following parameters are visible, if the parameter *distinction between long and short operation* has been set to the value *yes*:



Connected contact type	normally closed normally open
Reaction on short operation	ON OFF TOGGLE no reaction
Reaction on long operation	ON OFF TOGGLE no reaction
Long operation after: time base	100 ms / 1 s / 10 s / 1 min / 10 min / 1h
Factor [2...255]	2…5…255
Number of objects for short/long operation	1 communication object 2 communication objects
Debounce time	10 ms…50 ms…150 ms debounce time

Distinction between long and short operation

If the parameter is set to *no,* the input will be evaluated normally on every edge of the input signal. Evaluation is undertaken immediately.

If *yes* is selected, there is a delay after opening/closing the contact to determine whether there is a short or long operation. Only thereafter will a possible reaction be triggered.

The following table shows the function in detail:

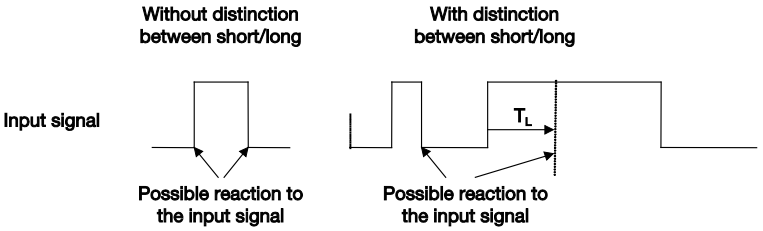


Fig. 3: Distinction between short/long operation for function “Switch”

T_L is the time duration from where a long operation is detected.

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Reaction on short operation
Reaction on long operation

For every operation (short or long) it is set if the object value is *ON, OFF* or *TOGGLE*, or if no *reaction* should be occur. The object value is updated as soon as it has been determined if a short or long operation has occurred.

Long operation after: time base, factor

Here the time period T_L after which an actuation is considered a “long” operation is set. (T_L = Time base x Factor).

Number of objects for short/long operation

In order to differentiate between long and short operation, a further communication object can be released by the option *2 communication objects*. This additional object reacts exclusively to long operations, where as the existing object reacts exclusively to short operation.

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

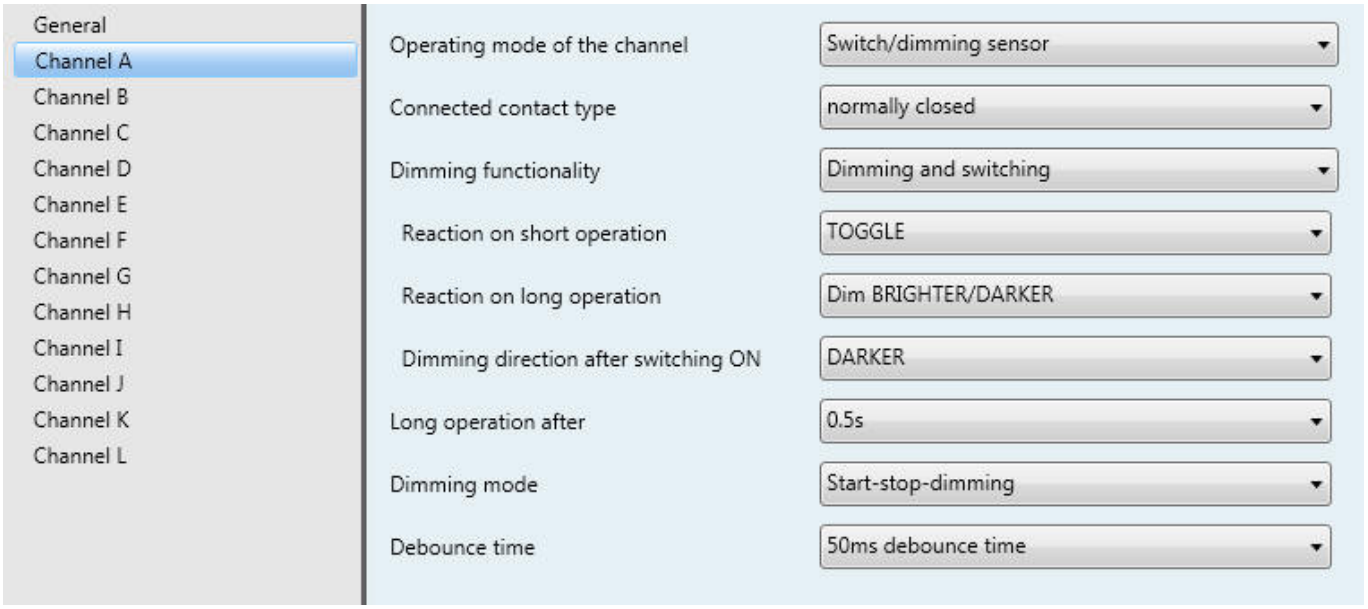
3.2.2.3 Communication objects “Switch sensor”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 bit** **DPT 1.003**	**C, W**
0: enable input 1: disable input The function of the input circuitry can be blocked or enabled using the communication object *Block*. A blocked input behaves as if there has been no change of the input signal. The input objects continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the state of the input has changed during blocking. If the input is just being operated as it is being enabled, the input behaves as if the operation has just commenced. If the input is just being operated as it is blocked, the response is undefined.				
1	**Switch**	**Channel A**	**1 bit** **DPT 1.001**	**C, W, T**
0: OFF 1: ON In accordance with the parameter setting, this communication object can be switched by actuation of the ON, OFF or TOGGLE input.				
2	**Switch**	**Channel A, long operation**	**1 bit** **DPT 1.001**	**C, W, T**
0: OFF 1: ON This object is only visible if the parameter *Distinction between long and short operation = yes*, and the parameter *Number of objects for short/long operation = 2 communication objects*. This additional object is only sent on long operation. If this object is visible, the object “Switch” only reacts with a short operation.				

3.2.3 Operating mode “Switch/Dimming sensor”

The operating mode “Switch/dimming sensor” is described in the following.

3.2.3.1 Parameters



Operating mode of the channel	Switch/dimming sensor
Connected contact type	normally closed normally open
Dimming functionality	Dimming and switching Only dimming
Reaction on short operation	ON OFF TOGGLE no reaction
Reaction on long operation	Dim BRIGHTER Dim DARKER Dim BRIGHTER/DARKER
Dimming direction after switching ON	BRIGHTER DARKER
Long operation after	0.3 s…0.5 s…10 s
Dimming mode	Start-stop-dimming Dimming steps
Debounce time	10 ms…50 ms…150 ms debounce time

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

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 the latter case, a long operation actuates dimming and a short operation actuates switching.

The advantage of the *Only dimming* function is that no distinction is made between short and long actuation. The dim command is issued immediately on actuation. It is not necessary to wait for a long actuation.

How does 1 button dimming function?

Switching and dimming functions can be fully controlled via a single push button. Each long dimming actuation is sent alternately with a BRIGHTER or DARKER dim telegram.

The “**1 button dimming”** is preset in the parameters. The function is as follows: If the communication object “Switch” = 0, a BRIGHTER telegram is sent at all times in case of a long operation. In order to evaluate the switch feedback of the actuator, the “Write” flag of the object “Switch” is set.

The following table shows the function in detail:

Value of the object “Switch”	**Value of the last dimming telegram**	**Reaction to dimming operation (sent dimming telegram)**
OFF	DARKER	BRIGHTER
OFF	BRIGHTER	BRIGHTER
ON	DARKER	BRIGHTER
ON	BRIGHTER	DARKER

Tab. 1: Dimming function “1 button dimming”

How does 2 button dimming function?

If “**2 button dimming**” is required, any two channels must be used for dimming with one for switch on / brighter and the other for switch off / darker.

The corresponding values must be set in the parameters *Reaction on short (or long) operation*: ON and Dim BRIGHTER for one button and OFF and Dim DARKER for the other button.

The objects “Switch” and “Dimming” of both channels are to be assigned with the same group addresses.

The user has the highest possible level of freedom using this solution.

Reaction on short operation

This parameter is visible if in the parameter *Dimming functionality* the value *Switch and Dimming* are set.

A short operation changes the value of the object *Switch*. This parameter sets if the object *Switch* toggles with short operation (typically: 1 button dimming) or only switches OFF or ON (typically: 2 button dimming).

Reaction on long operation

This parameter is visible if in the parameter *Dimming functionality* the value *Switch and Dimming* are set. A long operation changes the value of the object *Dimming*.

This parameter sets whether the object *Dimming* sends a BRIGHTER or a DARKER telegram after a long operation. The paramete “Dim BRIGHTER/DARKER” must be set for dimming with 1 button. In this case the dimming command is sent which is diametrically opposed to the last dim command.

Dimming direction after switching ON

In this parameter, you can be set whether the lighting should dim BRIGHTER or DARKER after switching on with the first long operation.

Example: If the brightness on switching on is only 20%, it makes sense to dim BRIGHTER after the lighting was switched on (parameter setting *BRIGHTER*).

Long operation after

This parameter is visible if in the parameter *Dimming functionality* the value *Switch and Dimming* is set. Here the time period $T_{L,}$ after which an actuation is considered a “long” operation, is defined.

Reaction on operation

This parameter is visible if the dimming function *Only dimming* has been set. No distinction is made between a short and long operation). The meaning of the adjustable settings corresponds to those of the parameter *Reaction on long operation* (see above).

Dimming mode

Start-stop-dimming is the standard dimming mode. It 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 operation.

Brightness change on every sent telegram

This parameter is only visible with the *Dimming steps* option. This parameter is set to change the brightness (in percent), which is cyclically sent with every dim telegram.

Sending cycle time: Telegram is repeated every

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.

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

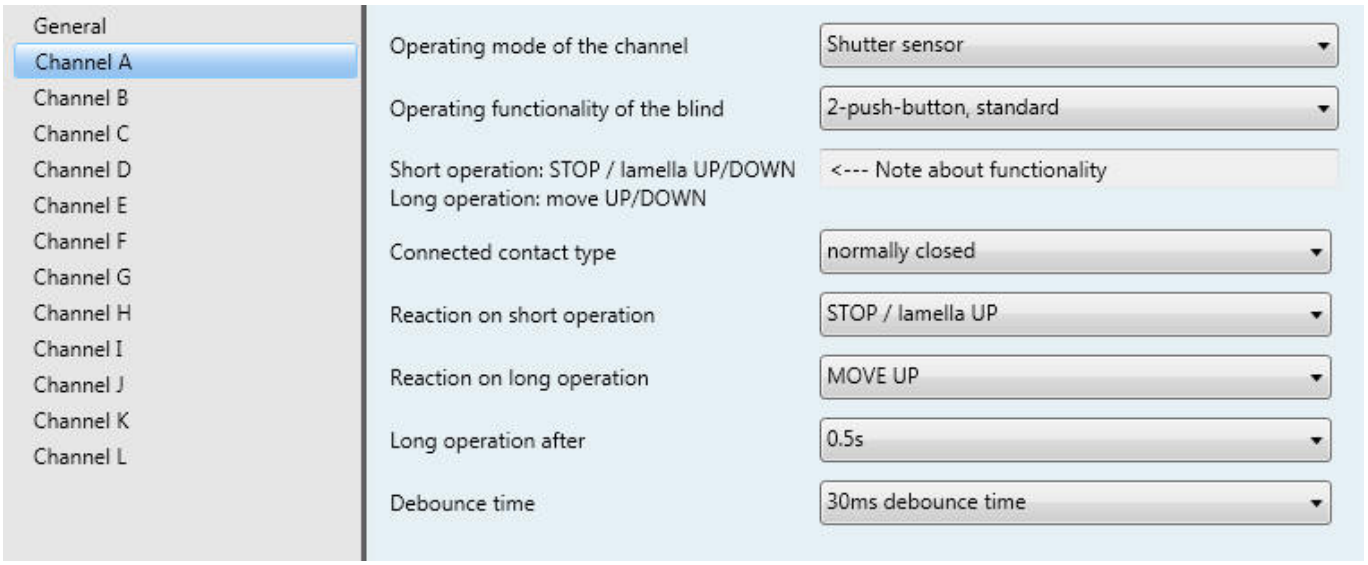
3.2.3.2 Communication objects “Switch/dimming sensor”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 bit** **DPT 1.003**	**C, W**
0: enable input 1: disable input The function of the input circuitry can be blocked or enabled using the communication object *Block*. A blocked input behaves as if there has been no change of the input signal. The input objects continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the state of the input has changed during blocking. If the input is just being operated as it is being enabled, the input behaves as if the operation has just commenced. If the input is just being operated as it is blocked, the response is undefined.				
1	**Switch**	**Channel A**	**1 bit** **DPT 1.001**	**C, W, T**
This object is visible if in the parameter *Dimming functionality* the value *Switch and Dimming* is set. The object value can be switched to ON, OFF or TOGGLE in accordance with the parameter setting with a short operation. With 1-switch dimming, the communication object as the non-sending group address should be linked with the switch feedback of the dimming actuator. Thus the input is informed via the current switching state of the dimming actuator.				
2	**Dimming**	**Channel A**	**4 bit** **DPT 3.007**	**C, T**
A long operation at the input has the effect that a *BRIGHTER* or *DARKER* dim command is sent via this communication object on the bus. A STOP command is sent at the end of actuation.				

3.2.4 Operating mode “Shutter Sensor”

The operating mode “Shutter sensor” is described in the following.

3.2.4.1 Parameters



Operating mode of the channel	Shutter sensor
Operating functionality of the blind	1-push-button, short = stepping, long = moving 1-push-button, short = moving, long = stepping 1-push-button-operation, moving 1-switch-operation, moving 2-push-button, standard 2-switch-operation, moving (shutter) 2-push-button, moving (shutter) 2-push-button, stepping
Connected contact type	normally closed normally open
Reaction on short operation	STOP / lamella UP STOP / lamella DOWN
Reaction on long operation	MOVE UP MOVE DOWN
Long operation after	0.3 s…0.5 s…10 s
Debounce time	10 ms…30 ms…150 ms debounce time

Operating functionality of the blind

These parameters define the type of operation. The following list provides an overview of the operating modes:

1-push-button, short = stepping, long = moving	
Short operation	STOP / lamella adjustment Opposite direction to the last movement command* To return to lamella adjustment, the blind must be raised or lowered briefly.
Long operation	Alternately “MOVE UP” or “MOVE DOWN”
1-push-button, short = moving, long = stepping	
Short operation	Alternately “MOVE UP” or “MOVE DOWN”
Long operation	STOP / lamella adjustment (cyclical sending); Opposite direction to the last movement or stepping command*.
1-push-button-operation, moving	
On operation	The following commands are sent in sequence: ... → “Move UP” → “Stop/Lamella adjustment UP” → “Move DOWN” → “Stop/Lamella adjustment DOWN” → ... *
1-switch-operation, moving	
Start of operation	Alternately “MOVE UP” or “MOVE DOWN”
End of operation	STOP / lamella adjustment *
2-push-button, standard	
Short operation	“Stop/Lamella adjustment UP” or “… DOWN” (programmable)
Long operation	“MOVE UP” or “MOVE DOWN” (programmable)
2-switch-operation, moving (shutter)	
Start of operation	“MOVE UP” or “MOVE DOWN” (programmable)
End of operation	“Stop/Lamella adjustment UP” or “… DOWN” (programmable)
2-push-button, moving (shutter)	
On operation	The following commands are sent in sequence: ... → MOVE UP STOP / lamella UP → ... or ....→ MOVE DOWN → STOP / lamella UP → ....
2-push-button, stepping	
On operation	“Stop/Lamella adjustment UP” or “…. DOWN“ is sent cyclically on the bus

* **Note:** If the actuator signals the upper limit position, (see object *Upper limit position*), it will move downwards with the next movement command. The same applies for the lower limit position.

In 1 push button/switch operation, the last direction of movement is determined via the last update of the object *Shutter UP/DOWN*.

How does the operation of a shutter function using a push button function?

The shutter function (movement and lamella adjustment) can be controlled completely using a single push button.

With operation via a normal push button normally “Short = lamella, long = moving“ (see above) is used. The operation is as follows:

With a long button push the lamella moves opposite to the last direction of movement. The user can stop movement with a short button push. Further short button pushes adjust the lamella against the last direction of movement.

What must be observed with the operation of a shutter using several separate push buttons?

In this case, the object “Shutter UP/DOWN“ and “STOP / lamella adjustment“ of the channels which are connected to the push buttons have each to be connected to the same group addresses.

Accordingly, a channel can “listen” to the commands of another channel. In this way, it always knows the last direction of movement.

What are the objects “Upper limit position” and “Lower limit position” for?

Using these objects, the Shutter Actuator informs if the shutter is in the upper or lower limit position. The Universal Interface then knows that the shutter has been moved to the upper limit position, for example, using a central command. The next movement command from a push button will always trigger a “downward” movement.

The latest generation of ABB shutter actuators support the objects “Upper limit position” and “Lower limit position”. If other shutter actuators are used, 1-button control is not recommended.

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Reaction on operation

This parameter is visible if there is no distinction between short and long actuation. It can be set whether the input triggers commands for movement upwards (*UP*) or downwards (*DOWN*).

Reaction on short operation
Reaction on long operation

This parameter is visible in operation if there is a distinction between short and long actuation. It can be set whether the input triggers commands for movement upwards (*UP*) or downwards (*DOWN*).

Long operation after

This parameter is visible in operation if there is a distinction between short and long actuation. Here the time period after which an actuation is considered a “long” operation is defined.

Telegram "Lamella" is repeated every …

This parameter is visible in operations in which the object *STOP / lamella adjustment*. is sent cyclically on the bus during long actuation. Here the time between two telegrams is set.

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

3.2.4.2 Communication objects “Shutter sensor”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 bit** **DPT 1.003**	**C, W**

The function of the input circuitry can be blocked or enabled using the communication object *Block*. A blocked input behaves as if there has been no change of the input signal. The input objects continue to be available.

0: enable input

1: disable input

When a disabled input is enabled, no telegrams are initially sent on the bus, even if the state of the input has changed during blocking. If the input is just being operated as it is being enabled, the input behaves as if the operation has just commenced.

If the input is just being operated as it is blocked, the response is undefined.

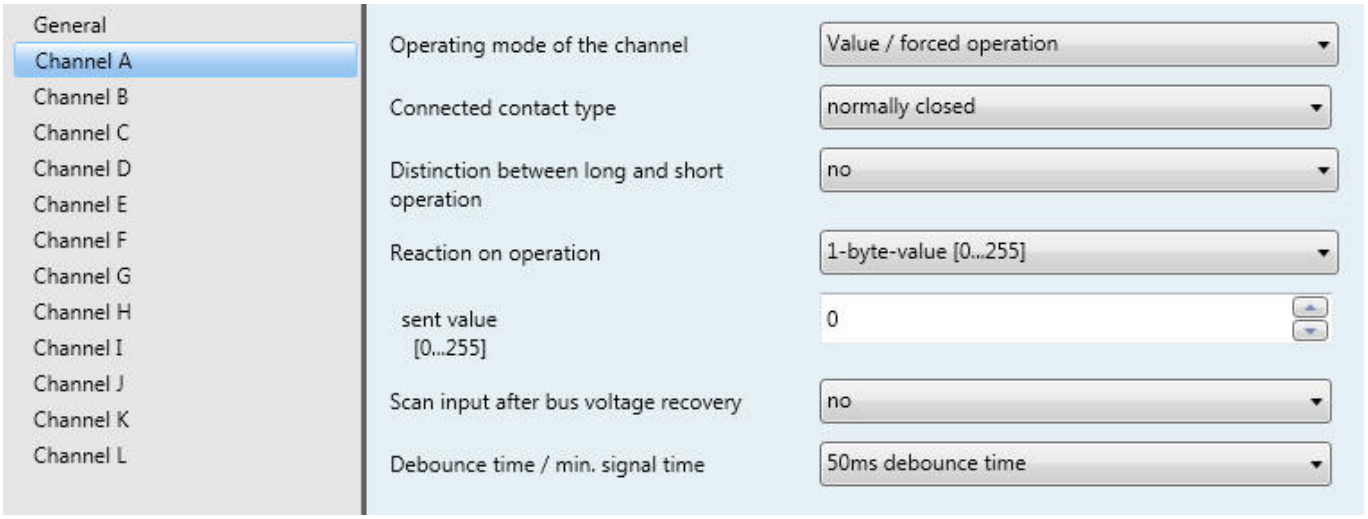
No.	Function	Object name	Data type	Flags
1	**Shutter UP/DOWN**	**Channel A**	**1 bit** **DPT 1.008**	**C, W, T**
This communication object sends a shutter motion command (UP or DOWN) on the bus. By receiving telegrams, the device also recognises movement commands of another sensor. 0: move upwards (UP) 1: move downwards (DOWN)				
2	**STOP/lamella adjustment**	**Channel A**	**1 bit** **DPT 1.007**	**C, T**
This communication object sends a STOP command or lamella adjustment. 0: STOP / lamella adjustment UP 1: STOP / lamella adjustment DOWN				
3	**Upper limit position**	**Channel A**	**1 bit** **DPT 1.002**	**C, W**
Using this object the shutter actuator indicates if it is in the upper limit position. The object is intended for 1-button operation. 0: Upper end limit not reached 1: Upper end limit reached **Note:** The communication object is important for 1-button operation.				
4	**Lower limit position**	**Channel A**	**1 bit** **DPT 1.002**	**C, W**
Using this object, the shutter actuator indicates if it is in the lower limit position. The object is intended for 1-button operation. 0: Lower end limit not reached 1: Lower end limit reached **Note:** The communication object is important for 1-button operation.				

3.2.5 Operating mode “Value / forced operation”

The operating mode “Value / Forced operation” is described in the following. The operating mode allows the sending of values of any data types.

3.2.5.1 Parameter “without short/long operation”

The following parameters are visible, if the parameter *distinction between long and short operation = no* has been set:



Operating mode of the channel	Value / forced operation
Connected contact type	normally closed normally open
Distinction between long and short operation	yes no
Reaction on operation	no reaction 2-bit-value (forced operation) 1-byte-value [0...255] 2-byte-value [-32768...32767] 2-byte-value [0...65535] 2-byte-value (floating point) 4-byte-value [0…4294967295]
sent value	Dependent on the selection made at *Reaction on operation*
Scan input after bus voltage recovery	yes no
Debounce time / min. signal time	10 ms…50 ms…150 ms debounce time Minimum signal time

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Distinction between long and short operation

In this parameter, you set if the input differentiates between short and long operation.

In the following, the parameters are described that are visible if there is no distinction between short and long actuation.

Reaction on operation

This parameter determines the data type that is sent when a contact is actuated.

Sent value

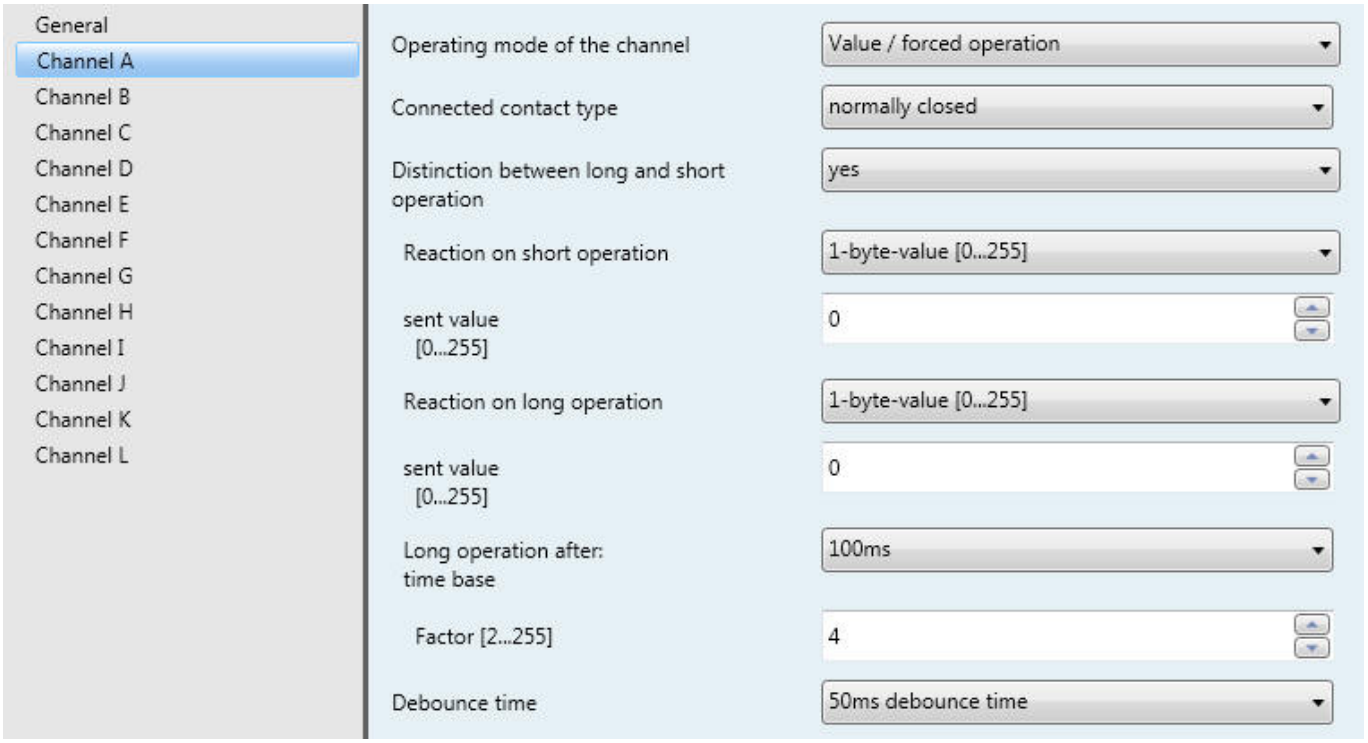
This parameter defines the value which is sent on actuation. The value range is dependent on the set data type.

Debounce time / min. signal time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

3.2.5.2 Parameter “with short and long operation”

The following parameters are visible, if the parameter *distinction between long and short operation = yes* has been set:



Reaction on short operation	no reaction 1-bit-value 2-bit-value (forced operation) 1-byte-value [0...255] 2-byte-value [-32768...32767] 2-byte-value [0...65535] 2-byte-value (floating point) 4-byte-value [0…4294967295]
sent value	Dependent on the selection made at *Reaction on operation*
Reaction on long operation	no reaction 1-bit-value 2-bit-value (forced operation) 1-byte-value [0...255] 2-byte-value [-32768...32767] 2-byte-value [0...65535] 2-byte-value (floating point) 4-byte-value [0…4294967295]
sent value	Dependent on the selection made at *Reaction on operation*

Long operation after: time base	100 ms / 1 s / 10 s / 1 min / 10 min / 1h
Factor [2...255]	2…4…255
Debounce time	10 ms…50 ms…150 ms debounce time

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Distinction between long and short operation

In this parameter, you set if the input differentiates between short and long operation. If *yes* is selected, there is a delay after operation to determine whether there is a short or long operation, and then the appropriate reaction follows.

Reaction on short operation
Reaction on long operation

This parameter determines the data type that is sent with a long or short operation.

Sent value

This parameter defines the value which is sent on short or long operation. The value range is dependent on the set data type.

Long operation after

Here the time period T_L after which an actuation is considered a “long” operation is defined.

T_L = Time base x Factor

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

3.2.5.3 Communication objects “Value/forced operation”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 bit** **DPT 1.003**	**C, W**

The function of the input circuitry can be blocked or enabled using the communication object *Block*. A blocked input behaves as if there has been no change of the input signal. The input objects continue to be available.

0: enable input

1: disable input

If the input is just being operated as it is blocked, the response is undefined.

When a disabled input is enabled, no telegrams are initially sent on the bus, even if the state of the input has changed during blocking. If the input is just being operated as it is being enabled, the input behaves as if the operation has just commenced at the end of the disable.

1	**Value (…)**	**Channel A**	**EIS variable** **DPT variable**	**C, T**

This communication object sends a value on the bus. The value and data type can be freely set in the parameters:

1 bit [0 / 1]	EIS 1	DPT 1.001	switch command
2 bit [0...3]	EIS 8	DPT 2.001	forced operation
1 byte [0...255]	EIS 6	DPT 5.010	brightness or position
2 byte [-32768...+32767]	EIS 10	DPT 7.001	signed value
2 byte [0...65535]	EIS 10	DPT 8.001	unsigned value
2 byte [floating point value*]	EIS 5	DPT 9.001	temperature
4 byte [0...4294967295]	EIS 11	DPT 12.001	value, unsigned

*sends values with fixed exponent of 3

1 **2**	**Value (…)** **Value (…)**	**Channel A, short operation** **Channel A, long operation**	**EIS variable** **DPT variable**	**C, T**

These communication objects send a value to the bus in case of long or short operation. The value and data type can be freely set in the parameters (see above).

Note: By default, the value objects of the “Write” flag are deleted (exception: 1 bot objects). Thus the object value cannot be changed via the bus. If this function is required, the “Write” flag must be set in the ETS. The object value is overwritten with the parameterised value after bus voltage recovery.

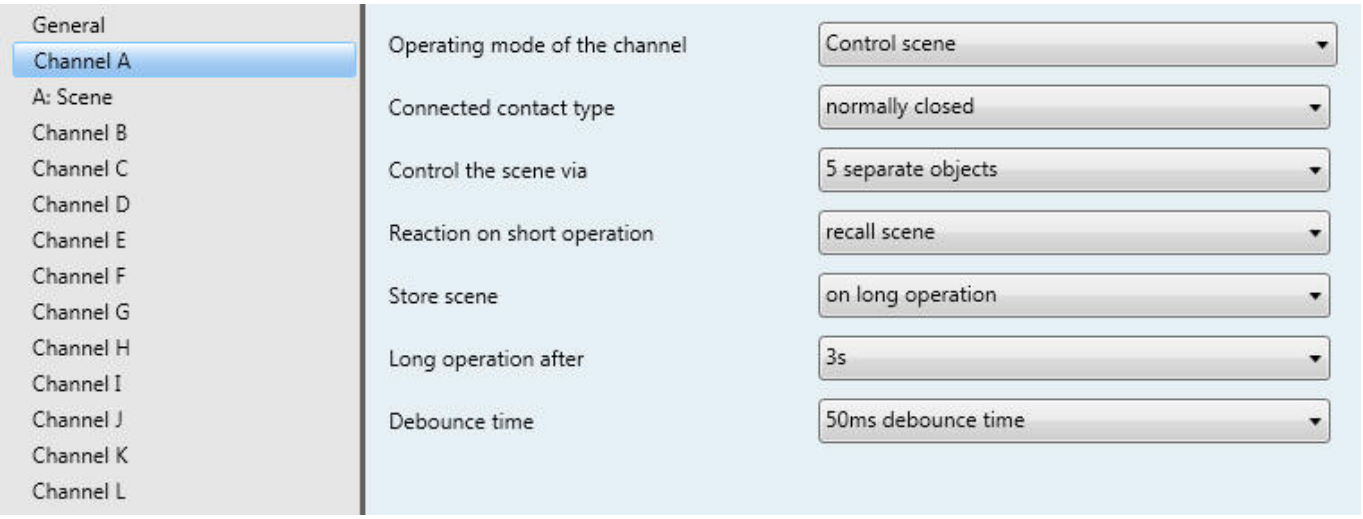
3.2.6 Operating mode “Control scene”

The operating mode “Control scene” is described in the following. It facilitates the recall and storing of states of several actuators or actuator groups.

A scene can be controlled via *5 separate objects* or *8 bit scene.*

3.2.6.1 Parameter with control via “5 separate objects”

This parameter window is visible when the scene is controlled via *5 separate objects.*



Operating mode of the channel	Control scene
Connected contact type	normally closed normally open
Control the scene via	5 separate objects
Reaction on short operation	no reaction Recall scene
Store scene	no on long operation with object value = 1 on long operation (if object value = 1)
Long operation after	0.3 s…3 s…10 s
Debounce time	10 ms…50 ms…150 ms debounce time

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Control the scene via

It is possible to select whether the scene control is carried out via *5 separate objects* or an *8 bit scene.* Parameters of the *8 bit scene* are described in the next section.

How does scene control via 5 separate objects function?

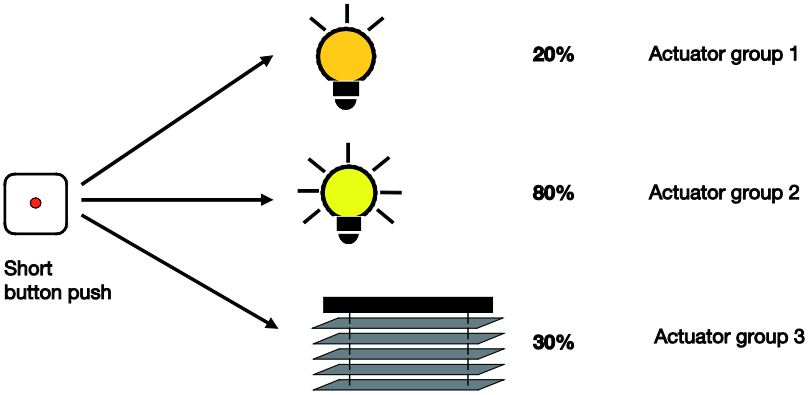


Fig. 1: Recall scene via “5 separate objects”

An input recalls up to 5 actuator groups. Up to 5 telegrams can thus be sent.

An actuator group comprises of one or several actuators that are linked with the same group address. It can consist, for example, of switch actuators (1 bit values) or dimming actuators (1 byte values).

How is a scene saved?

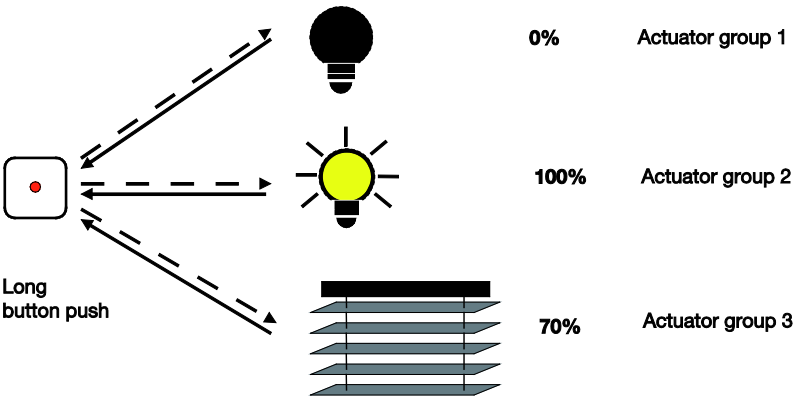


Fig. 2: Storing scene via 5 separate objects

The storing of a scene is carried out with a long push button action. The device scans each individual actuator group for the current value and then stores this value as the new scene value.

Please note that the “Read” flag has to be set for at least one actuator in each actuator group.

How does the user operate a scene?

The scene is controlled in the following way: A short operation calls a scene where as a long operation stores a scene.

What happens on a bus voltage failure?

On bus voltage failure, the scenes are reset to the parameterised values. Scenes that have been stored by the user will be lost.

Reaction on short operation

This parameter indicates whether a light scene is recalled after a short operation of the input or whether there is no reaction.

Store scene

This parameter defines in which way the storing of the current scene can be triggered as well as the functionality of the object *Store scene.* This is depedendent on the control of the scene.

Parameter value	Reaction
“on long operation”	On a long operation, the object values *Actuator group A…E:* s*witch* or *value* are read out via the bus and stored as new scene values. At the same time, the object “Store scene” sends the value “1” on the bus. The object value is reset to “0” when the button is released. It can thus be used to signal successful storage to the user.
“with object value = 1”	If the object “Store scene” receives the value “1”, the object values *Actuator group A..E switch* or *value* are read out via the bus. On receipt of the object value “0”, the current object values of Actuator groups A…E are stored permanently in the device. **Important:** **Storage of the current scene thus requires the object values “1” and “0” to be sent in succession!**
“on long operation AND object value = 1”	If the object “Store scene” receives the value “1” on the bus, the next long push button action leads to the scanning of the object values *Actuator group A..E switch* or *…value*. These values are stored as the new scene values. At the same time, with a long operation the object “Store scene” sends the value “1” on the bus. The object value is reset to “0” when the button is released. A long operation will be interpreted like a short operation if, the object value *Store scene* has not the value “1”, i.e. the scene will thus be generally recalled.

Long operation after

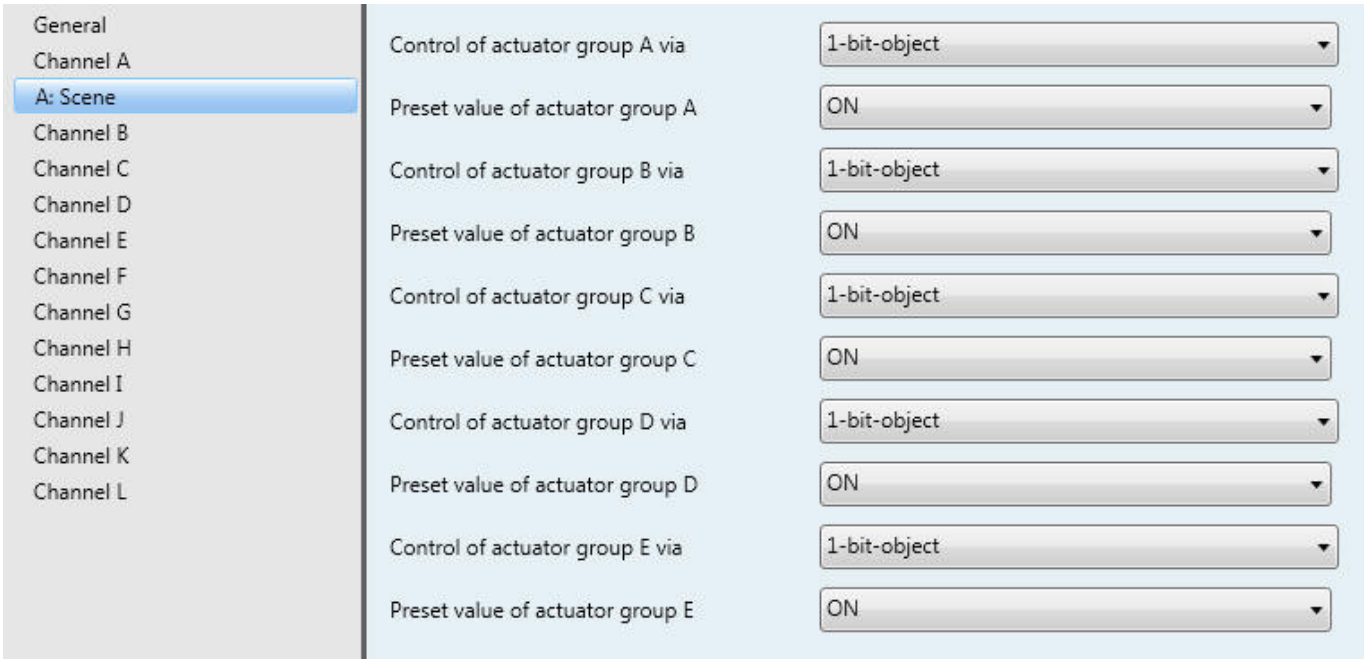
This parameter is visible if the storing of the scene is possible via a long operation. Here the time period after which an actuation is considered a “long” operation is set.

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

Parameter window “A: Scene”

This parameter window is visible if a light scene control via *5 separate objects* is selected.



Control of actuator group X via	1-bit-object 8-bit-object
Preset value of actuator group X	ON OFF

Control of actuator group A via
…
Control of actuator group E via

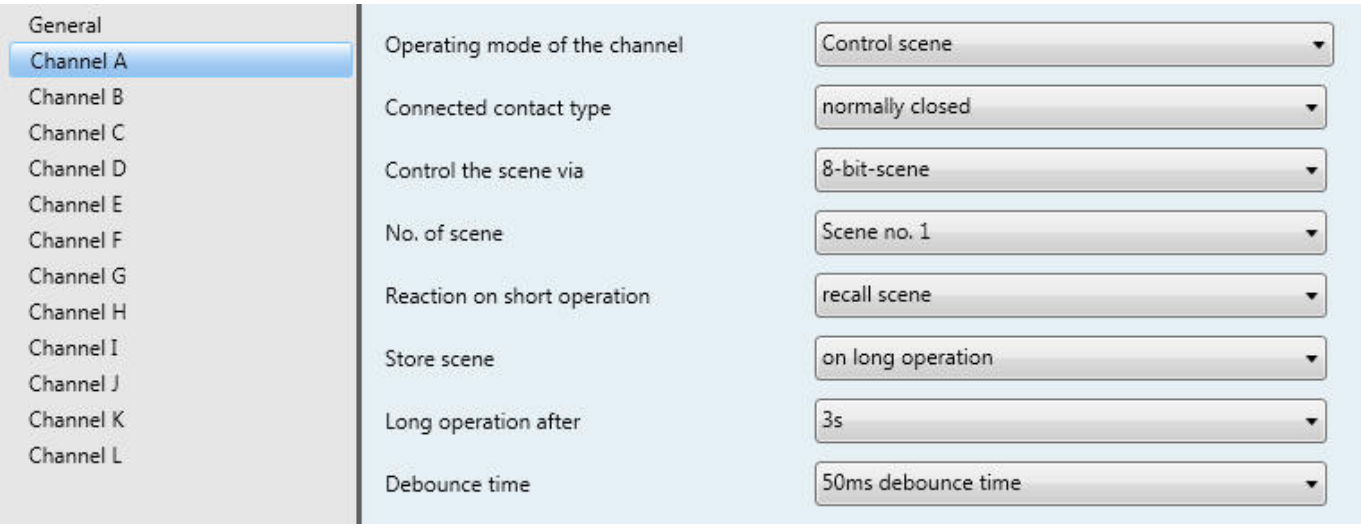
A value can be preset for each actuator group whether a *1-bit-object* or an *8-bit-object*. The type of communication object *Actuator group A to …E* is set accordingly.

Preset value of actuator group A
...
Preset value of actuator group E

In this parameter, a value can be preset for every actuator group A…E. If a scene has been stored, after programming or bus voltage recovery, the current object values of the actuator groups A…E are overwritten with the values set here.

3.2.6.2 Parameter with control via “8 bit scene”

This parameter window is visible when the scene is controlled via an *8 bit scene.*



Control the scene via	8 bit scene
No. of scene	Scene no. 1 … Scene no. 64
Reaction on short operation	no reaction Recall scene
Store scene	no on long operation with object value = 1 on long operation (if object value = 1)
Long operation after	0.3 s…3 s…10 s
Debounce time	10 ms…50 ms…150 ms debounce time

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Control the scene via

It is possible to select whether the scene control is carried out or stored in the actuators via *5 separate objects* or an *8 bit scene.* The parameters of the scene via *5 separate objects* are described in the previous section.

How does scene control via 8 bit scene?

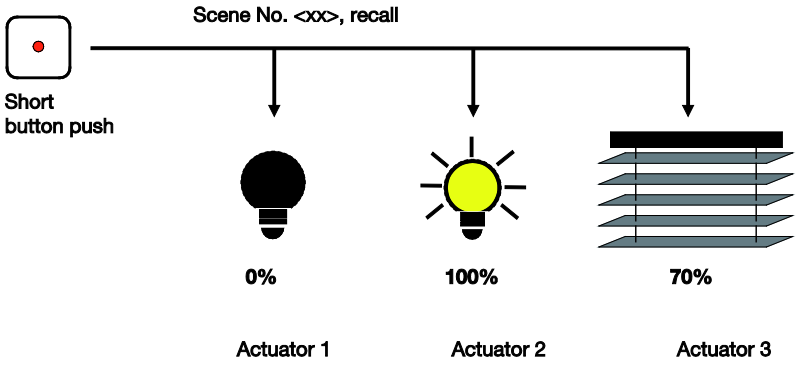


Fig. 3: Recalling and storing via “8 bit scene”

With the 8 bit scene, the push button issues the instruction to recall a scene. The scene is not stored in the push button but rather in the actuator. All actuators are addressed using the same group address. It is thus sufficient to send a single telegram to recall the scene.

A scene number is sent with the telegram value, which must correspond with the scene number in the parameters of the actuator.

Up to 64 different scenes are managed via a single group address. An 8 bit scene telegram contains the following information:

- No. of scene (1...64)

- Recall scene / Store scene

After a long push button action, the actuators receive a store command which causes them to store the current value as a new scene value.

What is the difference to a scene via 5 separate objects?

The main difference is that the scene values in the 8-bit scene are stored in the actuators. This requires actuators which support this function.

How does the user operate a scene?

The scene is controlled in the following way: A short operation calls a scene where as a long operation stores a scene.

No. of scene

The scene number (1...64) assigned to this channel is defined here.

Reaction on short operation

This parameter indicates whether a light scene is recalled after a short operation or whether there is no reaction.

Store scene

This parameter defines in which way the storing of the current scene can be triggered as well as the functionality of the object *Store scene.* This is dependent on the control of the scene.

Parameter value	Reaction
“on long operation”	On a long button operation, the object *8 bit scene* sends a storae command on the bus. This triggers the actuator to store the current scene in the actuators. The object *Store scene* has no function here.
“with object value = 1”	The object “8 bit scene” will send a store command on the bus, if the object “Store scene” receives the value “1”.
“on long operation AND object value = 1”	A long button operation triggers a store command via object *8 bit-scene* only if the object *Store scene* has the value “1“. A long operation will be interpreted like a short operation if, the object value *Store scene* has the value “0” or if no “1” has been received on this object since the last long button operation. This means that generally the scene will be recalled. The same applies for the case where the value “0” has been last received.

Long operation after

This parameter is visible if storing of the scene is possible via a long operation. Here the time period after which an actuation is considered a “long” operation is set.

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

3.2.6.3 Communication objects “Control scene”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 bit** **DPT 1.003**	**C, W**
The function of the input circuitry can be blocked or enabled using the communication object *Block*. A blocked input behaves as if there has been no change of the input signal. The input objects continue to be available. 0: enable input 1: disable input If the input is just being operated as it is blocked, the response is undefined. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the state of the input has changed during blocking. If the input is just being operated as it is being enabled, the input behaves as if the operation has just commenced.				
1 **...** **5**	**Actuator group A: switch** **...** **Actuator group E: switch**	**Channel A** **...** **Channel A**	**1 bit** **DPT 1.001**	**C, W, T**
1 **...** **5**	**Actuator group A: value** **...** **Actuator group E: value**	**Channel A** **...** **Channel A**	**1 byte** **DPT 5.010**	**C, W, T**
These objects are visible if the scene is controlled via *5 separate objects.* They control up to 5 actuator groups, either via 1 bit or 8 bit values (can be parameterised). When storing the scene, the device reads out the current value via the bus and stores it in these objects. On bus voltage recovery, the object values are overwritten with the parameterised values.				

No.	Function	Object name	Data type	Flags
1	**8 bit scene**	**Channel A**	**1 byte** **DPT 18.001**	**C, T**

This object is visible if the control is carried out via an “8 bit scene”.

It sends a scene number and the information as to whether a scene should be recalled or the current scene should be stored. The storing of the scene is carried out in the actuator.

Telegram code in bits: MxSSSSSS

(MSB) (LSB)

M: 0 – Scene is recalled

1 – Scene is stored

x: Not used

Number of scene (0...63 according scene no. 1...64)

A table of the object values can be found in section 6.2.

No.	Function	Object name	Data type	Flags
6	**Store scene**	**Channel A**	**1 bit** **DPT 1.003**	**C, W, T**

This object can be used to trigger the storing of a scene via the bus or to indicate that the scene has been stored. The function of the object depends the parameter “*Store scene”.*

When the object receives a telegram is has the following function:

0: Complete the storage of the current scene

1: Commence the storage of the current scene

If the object sends a telegram, it has the following function:

0: Storage of the current scene is completed

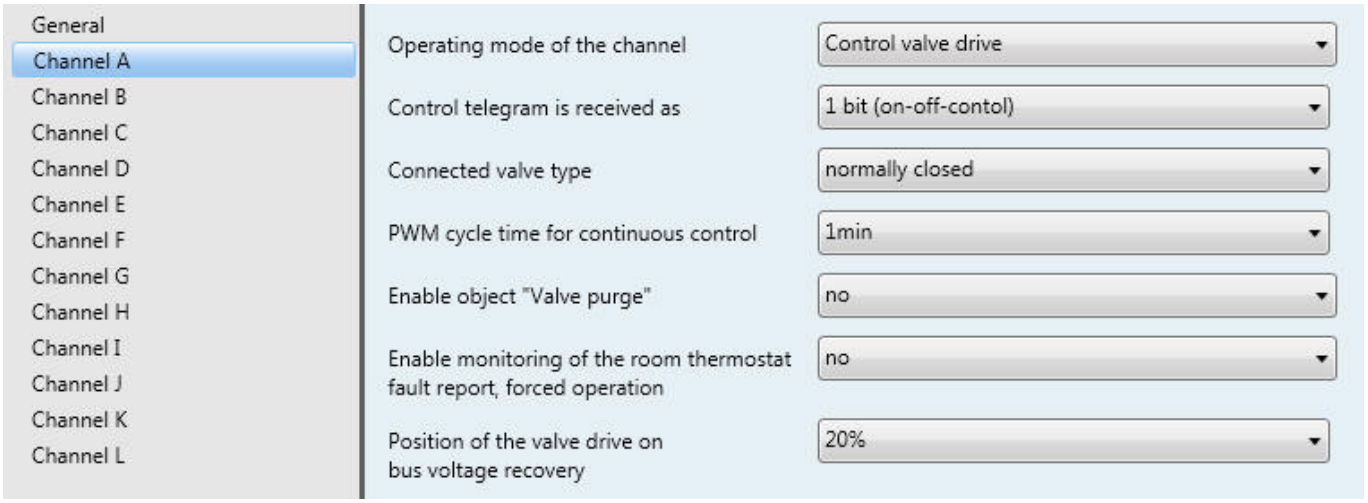
1: Storage of a scene was has been started

The object can be used simultaneously to trigger (received group address) and display (sending group address) storage of a scene via the bus.

3.2.7 Operating mode “Control valve drive”

The operating mode “Control valve drive” is described in the following. This operating mode controls the Universal Interface to which an electronic relay and an electro-thermal valve drive can be connected.

3.2.7.1 Parameters



Operating mode of the channel	Control valve drive
Control telegram is received as	1 bit (on-off contol) 1 byte (continuous control)
Connected valve type	normally closed normally open
PWM cycle time for continuous control	20 s…50 s…1 min…50 min…1 h
Enable object "Valve purge"	yes no
Enable monitoring of the room thermostat fault report, forced operation	yes no
Position of the valve drive on bus voltage recovery	0% (closed) … 20 % … 100% (open)

Control telegram is received as

The heating actuator can either be controlled via the 1 bit object “Switch” or the 1 byte object “Control value (PWM)”.

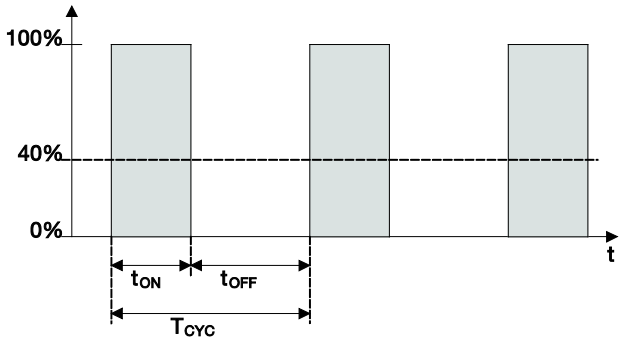
In the case of **1 bit** control, the heating actuator functions in a similar way to a normal switch actuator: The room thermostat sends normal switching commands (ON and OFF). Simple 2-step closed-loop control is usually implemented in this way.

For **1 byte** control, a value of 0…255 (corresponds to 0 %…100 %) is preset by the room thermostat. This process is also known as “continuous-action control”. At 0 %, the valve is closed and at 100 % it is fully opened. The heating actuator controls intermediate values via pulse width modulation with this method of control (see diagram below).

What does “Control via pulse width modulation (PWM)” mean

Pulse width modulation occurs when the room thermostat sends a “1 byte value” (value range 0…255) as a control value to the Universal Interface. This converts the value to a clocked (pulse width modulated) output signal.

With pulse width modulation, the control is implemented by a variable mark-space ratio. The following example makes this clear:



During t_{ON}, the valve is triggered with OPEN (“ON phase”). During t_{OFF} the valve is triggered with CLOSE (“OFF phase”). If t_{ON} = 0.4 x T_{CYC}, the valve is set at approx. 40%. T_{CYC} is the so-called PWM cycle time for continuous control.

However…

Even when the room thermostat sends “1 bit values” (switch commands) as control values to the Universal Interface, a pulse width modulated signal can be generated by quickly switching on and off. Due to the heavy bus load created by quick successive switching of ON and OFF telegrams, this process is however unusual.

Connected valve type

In this parameter, you can set whether a valve should be controlled as “*normally closed*” or “*normally open*”.

normally closed:

The valve closes when the electronic relay is opened.

normally open:

The valve closes when the electronic relay is closed.

PWM cycle time for continuous control

When 1 byte control is selected, this parameter sets the PWM cycle time $T_{CYC,}$ which is used to time the control signal.

If 1 bit control has been parameterized, this parameter is only used in fault mode, during forced positioning and directly after bus voltage recovery.

Enable object "Valve purge"

The object *valve purge* is enabled with this parameter.

Enable monitoring of the room thermostat fault report, forced operation

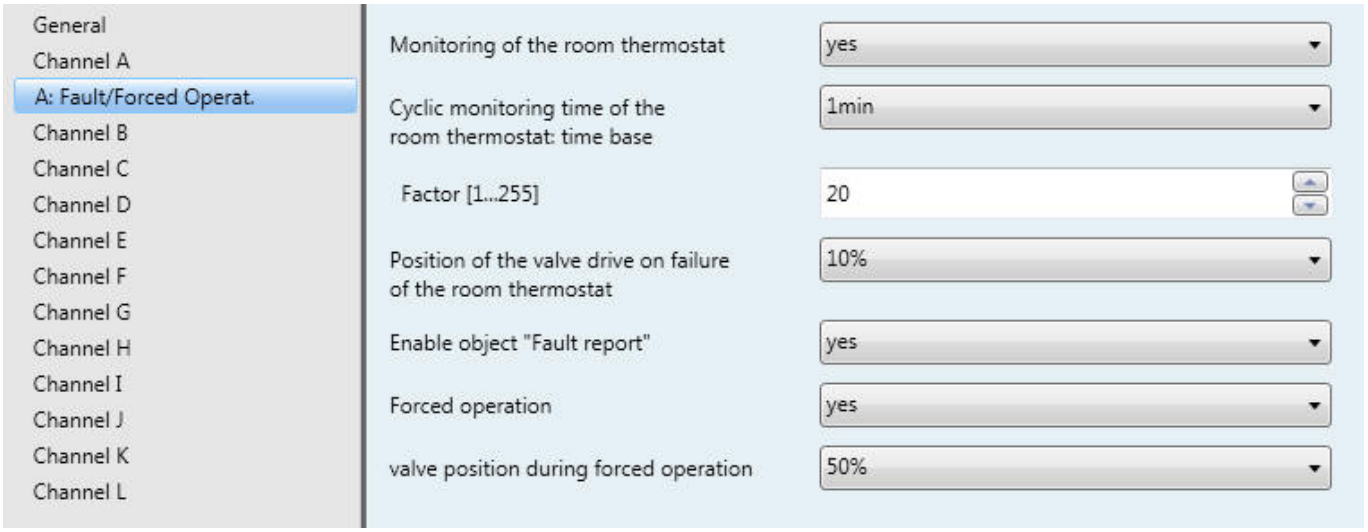
The parameter window *A: Fault/Forced Operat.* is enabled with this parameter. Further settings can be carried out in this window for the cyclical monitoring of the room thermostat and for the forced positioning of the actuator.

Position of the valve drive on bus voltage recovery

This parameter sets how the valve drive is triggered after bus voltage recovery until the first switching or positioning command is received from the room thermostat. The position is controlled via a PWM signal.

3.2.7.2 Parameter window “A-Fault/Forced Operat.”

This parameter window is visible if in parameter *Enable monitoring of the room thermostat fault report, forced operation = yes* is set.



Monitoring of the room thermostat	yes no
Cyclic monitoring time of the room thermostat: time base	1 s / 10 s / 1 min / 10 min / 1 h
Factor [1…255]	1…20…255
Position of the valve drive on failure of the room thermostat	0% (closed) … 10 % … 100% (open)
Enable object "Fault report"	yes no
Forced operation	yes no
valve position during forced operation	0% (closed) … 50 % … 100% (open)

Monitoring of the room thermostat

Using this parameter, cyclic monitoring of the thermostat is enabled.

The telegrams of the room thermostat are transferred at specific intervals. If one or more of the consecutive telegrams is omitted, this can indicate a communications fault or a malfunction in the room thermostat.

If the device for the **Cyclic monitoring time** does not receive a telegram on the object *Switch* or *Control value PWM*, the actuator switches to fault mode and triggers a safety position. The fault mode is ended as soon as a telegram is received.

Cyclic monitoring time of room thermostat

The cyclical monitoring of the room thermostat telegrams is enabled with this parameter.

Duration = Time base x Factor.

Position of the valve drive on failure of the room thermostat

This parameter defines the safety position which the device triggers in fault mode. The PWM cycle time T_{CYC} of the control is defined in the parameter *PWM cycle time for continuous control*.

Enable object “Thermostat fault”

The object *Thermostat fault* can be enabled in this parameter. It has the object value "ON" during a malfunction. If there is no fault, it has the object value “OFF”. The object is always sent cyclically. The cyclic transmission time is identical to the cyclic monitoring time.

Forced operation

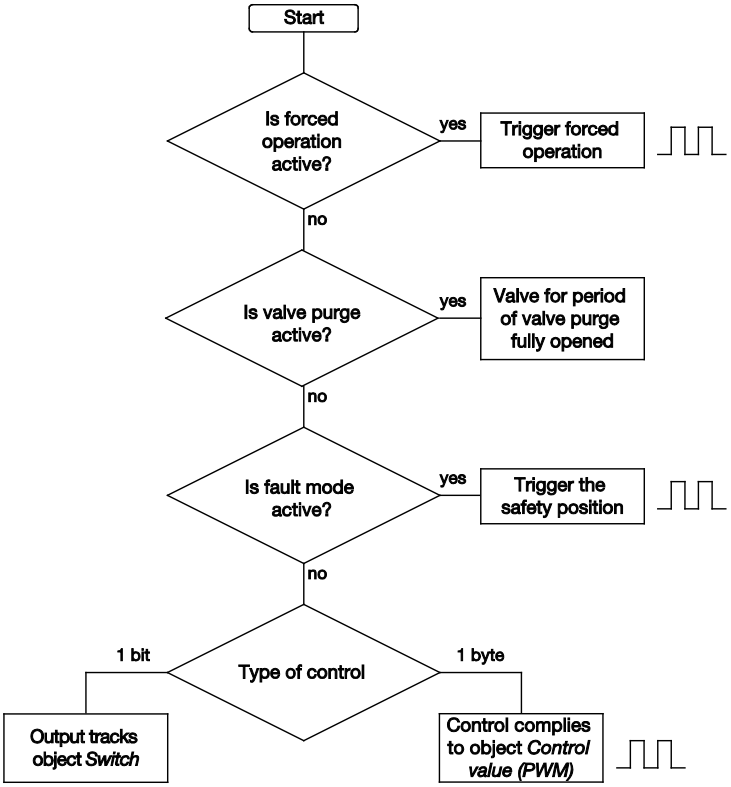
This parameter enables the function “Forced operation”. During a forced operation, the device triggers a freely adjustable forced position. This has the highest priority, i.e. it is not modified by a valve purge or safety position. Forced operation can be activated or deactivated via the object *Forced positioning*.

Valve position during forced operation

In this parameter, the valve position during the forced operation is defined. The PWM cycle time T_{CYC} of the control is defined in the parameter *PWM cycle time for continuous control*.

What are relationships between the functions “Forced operation”, “Valve purge” and “Fault of room thermostat”?

The following diagram provides an overview:



Forced positioning has the highest priority, followed by valve purging and fault mode operation.

How quick is the triggering of the special positions?

To improve the control behaviour with pulse width modulation (PWM), the special positions are sometimes not started or finished immediately but only once a PWM cycle time has elapsed or after an ON or OFF phase within the cycle.

The following table provides an overview:

Control of the valve via	**Reaction on start**	**Reaction on end**
Forced positioning	Control immediately	Once an ON or OFF phase has elapsed
Valve purge	Control immediately	End immediately
Fault mode	Once the cycle has elapsed	Once the cycle has elapsed

3.2.7.3 Communication objects “Control valve drive”

No.	Function	Object name	Data type	Flags
1	**Switch**	**Channel A**	**1 bit** **DPT 1.001**	**C, W**
This object is visible if the control by the room thermostat is carried out via a 1 bit object. If the object has the value “ON”, the valve is opened while the valve is closed if the object has the value “OFF”. 0: Valve closes 1: Valve opens				
1	**Control value (PWM)**	**Channel A**	**8 bit** **DPT 5.010**	**C, W**
0: Close valve fully ... Center position (Mark-to-space ratio) 255: Open valve fully This object is visible if the control of the heating actuator is implemented via an 8 bit object, e.g. within a continuous control. The object value [0…255] is determined by the variable mark-to-space of the valve.				
3	**Valve purge**	**Channel A**	**1 bit** **DPT 1.001**	**C, W**
0: End valve purge 1: Start valve purge This object is visible if the parameter *Enable object "Valve purge"* has the value *yes.* The valve purge of the device is activated or deactivated via this object. During the valve purge, the valve is controlled with “Open”.				

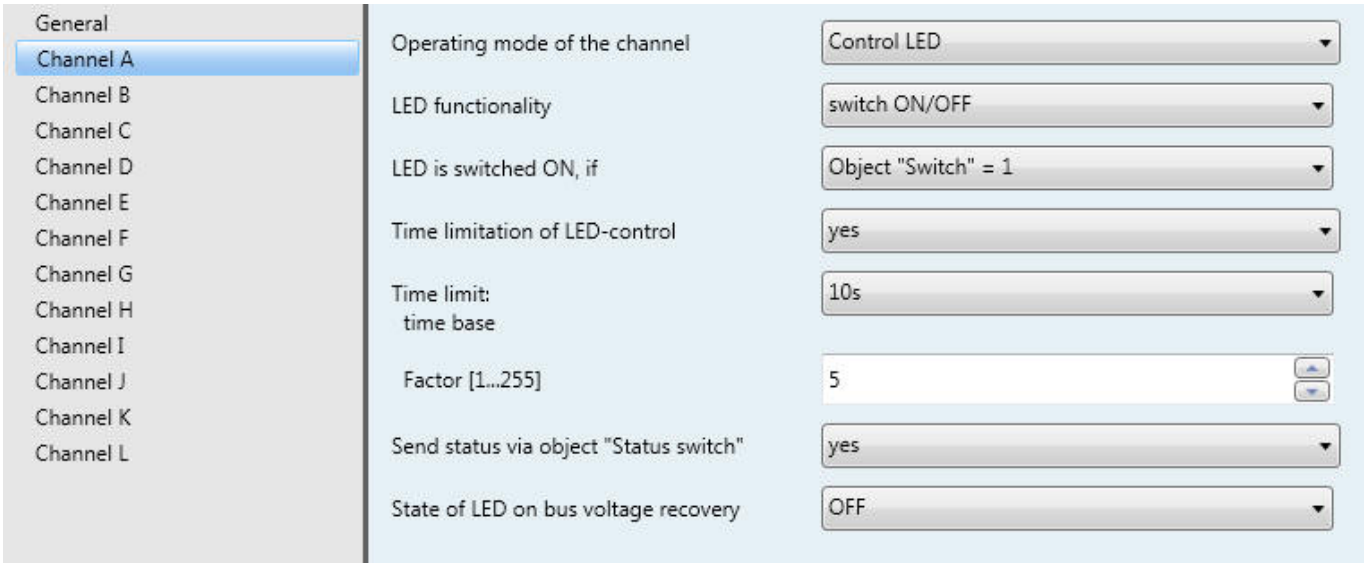
No.	Function	Object name	Data type	Flags
4	**Forced operation**	**Channel A**	**1 bit** **DPT 1.001**	**C, W**
0 = end forced operation 1 = start forced operation This object is visible if the 1 bit forced operation is enabled in the parameters. The forced operation of the device is activated or deactivated via this object. In this way, the valve can be controlled with a defined value. Forced operation has the highest priority.				
5	**Status switch**	**Channel A**	**1 bit** **DPT 1.001**	**C, T**
0: Valve closes 1: Valve opens This object reports the switching state of the heating actuator. The object value is sent after each change of the output. **Important:** **For PWM continuous control, this object is sent after each change in the output. The additional telegram load should therefore be taken into account!**				
6	**Fault report**	**Channel A**	**1 bit** **DPT 1.001**	**C, T**
0: No fault 1: Fault mode active This object is visible if the fault messages are enabled in the parameters. If the output does not receive any telegrams from the room thermostat via the object “Switch” or “Control value (PWM)” for an adjustable period, the devices switches to fault mode and reports this via the object.				

3.2.8 Operating mode “Control LED”

The operating mode “Control LED” is described in the following.

3.2.8.1 Parameters

Parameter window for *LED function = switch ON/OFF:*



Operating mode of the channel	Control LED
LED functionality	switch ON/OFF Flashing
LED is switched ON, if	Object "Switch" = 1 Object "Switch" = 0
Time limitation of LED-control	yes no
Time limit: time base	1 s / 10 s / 1 min / 10 min / 1 h
Factor [1…255]	1…5…255
Send status via object "Status switch"	yes no
State of LED on bus voltage recovery	OFF ON

LED functionality

This parameter defines whether the output should control the LED permanently (“switch ON/OFF”) or whether it should be “Flashing”. The corresponding objects “LED switch” or “LED flashing” are enabled.

In the following, the parameters for the *switch ON/OFF* setting are described.

LED is switched ON, if

It can be defined in which state of the object *LED switch* the LED is switched on.

Time limitation of LED-control

If *yes* has been entered in this parameter, the operating time of the LED has a time restriction.

Time limit (Time base/Factor)

If the time limit is active, it is possible to indicate in this parameter the maximum period that an LED is switched on. Once this time limit has elapsed, the LED is switched off.

Duration = Time base x Factor.

Send status via object "Status switch"

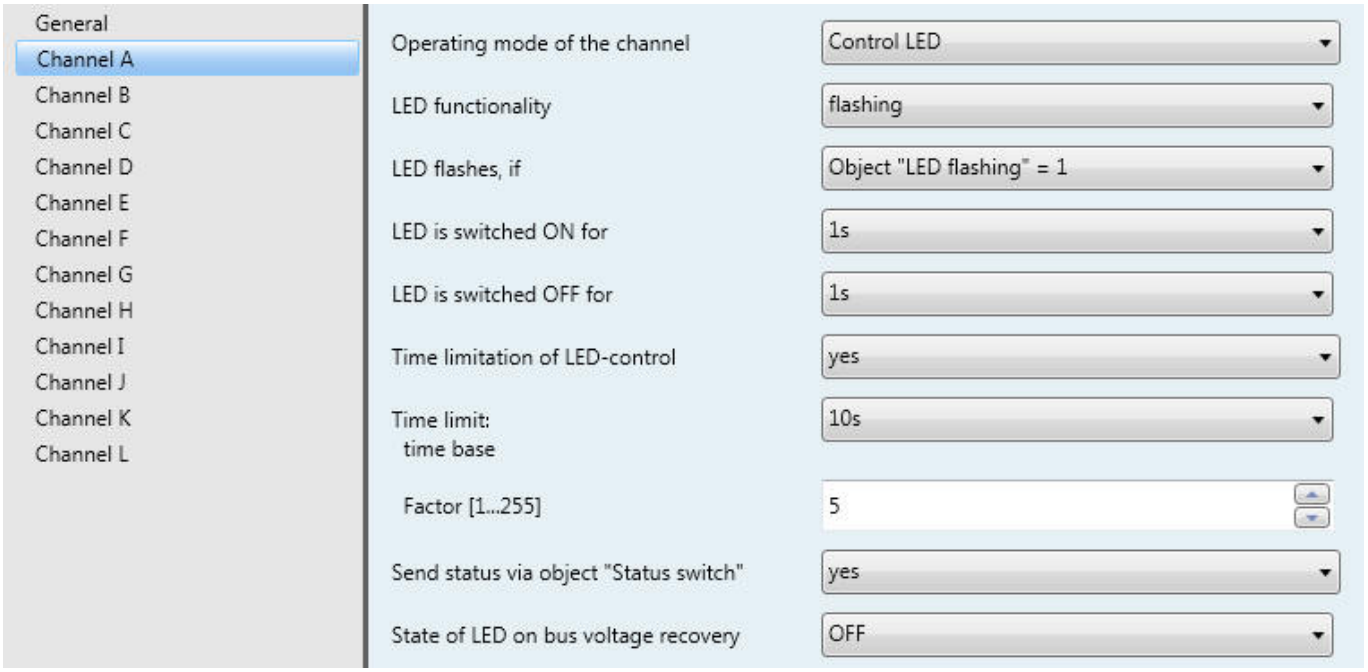
The object “Status switch” is enabled with this object. The object value is ON if the LED has been switched on.

State of LED on bus voltage recovery

Here you can be set whether the LED is switched ON or OFF after bus voltage recovery.

3.2.8.2 Parameters of LED function “Flashing”

Parameter window for *LED function = Flashing:*



LED flashes, if	Object "LED flashing" = 1 Object "LED flashing" = 0
LED is switched ON for	200 ms…800 ms…1 s…60 s
LED is switched OFF for	200 ms…800 ms…1 s…60 s
Time limitation of LED-control	yes no
Time limit: time base	1 s / 10 s / 1 min / 10 min / 1 h
Factor [1…255]	1…5…255
Send status via object "Status switch"	yes no
State of LED on bus voltage recovery	OFF ON

LED flashes, if

It can be defined which state the object *LED flashing* must have so that the flashing is active.

LED is switched ON for
LED is switched OFF for

The length of time for which the LED is switched on or switched off during the flashing signal is set. The flash rate of the signal can thus be set.

Time limitation of LED-control

If *yes* has been entered in this parameter, the flashing period of the LED has a time restriction.

Time limit (Time base/Factor)

If the time limit is active, it is possible to indicate in this parameter the maximum period for which an LED flashes. In this way, the number of flashing pulses can be limited. Once this time limit has elapsed, the LED is switched off.

Duration = Time base x Factor.

Send status via object "Status switch"

The object *Status switch* is enabled with this object. The object value is ON if the LED flashes.

State of LED on bus voltage recovery

Here you can be set whether the LED flashes (*ON*) or does not flash (*OFF)* after bus voltage failure.

3.2.8.3 Communication objects “Control LED”

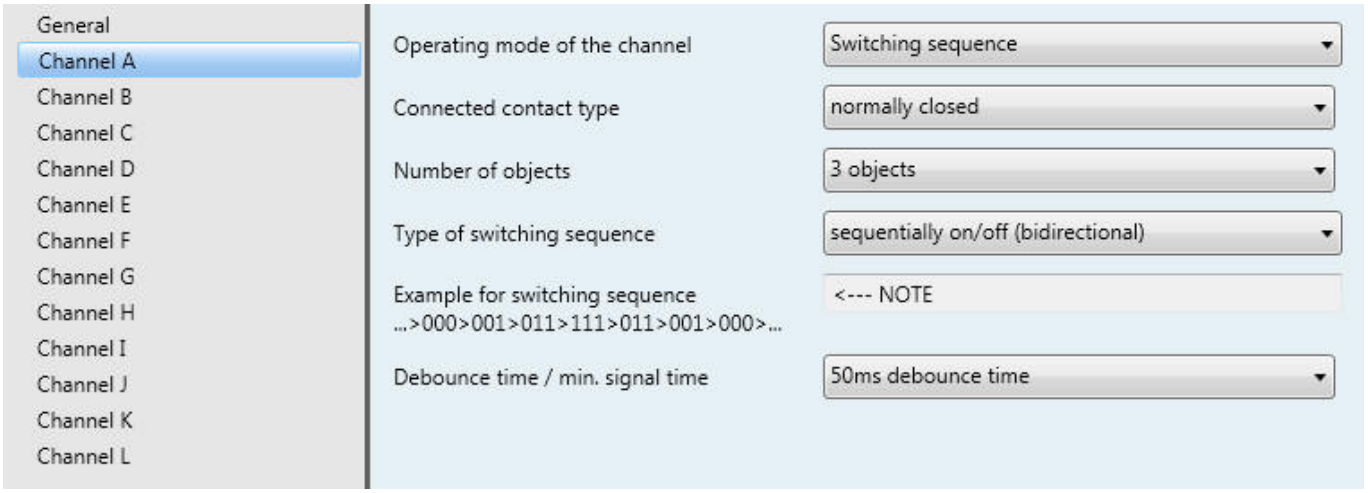
No.	Function	Object name	Data type	Flags
1	**LED switching**	**Channel A**	**1 bit** **DPT 1.001**	**C, W**
This object is visible if the parameter *LED function* = *Switch*. The object switches the LED ON and OFF. The telegram values can be set in the parameters.				
2	**LED flashing**	**Channel A**	**1 bit** **DPT 1.001**	**C, W**
This object is visible if the parameter *LED function* = *Flashing* is set. The flashing of the LED can be started and stopped via this object. 0: Stop flashing 1: Start flashing				
3	**LED permanent ON**	**Channel A**	**1 bit** **DPT 1.001**	**C, W**
This object is visible if the parameter *LED function* = *Flashing* is set. The LED can be switched on permanently via this object. For example, the flashing function is deactivated in this way. 0: Flashing function active 1: LED continuous ON				
4	**Status switch**	**Channel A**	**1 bit** **DPT 1.001**	**C, T**
This object is visible if the parameter *Send status via object* the value *yes* is set. The state of the output is fedback. 0: LED is switched off 1: LED is switched on or flashes				

3.2.9 Operation mode “Switching sequence”

The operating mode “Switching sequence” is described in the following.

It enables the modification of several object values in a defined switching sequence using a single push button.

3.2.9.1 Parameters



Operating mode of the channel	Switching sequence
Connected contact type	normally closed normally open
Number of objects	2 / 3 / 4 / 5 objects
Type of switching sequence	sequentially on/off (bidirectional) sequentially on/off (one direction) All combinations
Debounce time / min. signal time	10 ms…50 ms…150 ms debounce time Minimum signal time

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Number of objects

Here the number of communication objects (max. 5) are defined, which are to be used in the switching sequence. The corresponding objects *Telegr. value 1* to *Telegr. value 5* are enabled.

Type of switching sequence

The switching sequence can be selected here. The following switching sequences are possible:

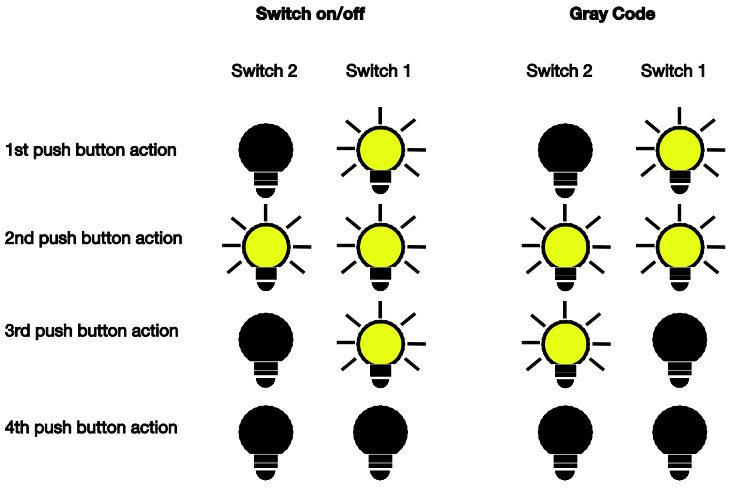
Type of switching sequence	Example
“sequentially on/off (bidirectional)”	...-000-001-011-111-011-001-...
“sequentially on/off (one direction)”	000-001-011-111
All combinations	...-000-001-011-010-110-111-101-100-...

The example relates to the state of three objects (“0” = OFF, “1” = ON). A table can be found in section Switching sequence “All combinations”, page 87.

What is the operating mode “switching sequence” used for?

The operating mode “Switching sequence” allows switch on or off of up to five objects (1 bit) in a defined sequence. Every time the button is switched one further step in the sequence occurs.

Example:



5th push button action like 1st

In this example, two lamps (or lamp groups) are controlled. Thus two objects are used.

How many lamps can be switched in a sequence?

Up to 5 lamps (or groups of lamps) can be switched.

Which switching sequences are available?

1) Sequentially on/off (one push-button)

This switching sequence switches on a further communication object with each successive actuation . If all communication objects are switched on, they are switched off successively commencing with the last one to be switched on.

2) Sequentially on/off, multiple push-buttons

The switching sequence is similar to the function “sequentially on/off (one push-button) with the exception that that you can only switch up or down via an input. If all the sequences reaches the end, all further operations in the same direction are ignored. At least two inputs are therefore required for this switching sequence.

3) All combinations

In this sequence all communication object combinations are undertaken successively. Only the value of a single communication object is changed each time. A good example of this switching sequence, for example, is switching of two lighting groups in the sequence

00 – 01 – 11 – 10 – 00 ...

A table can be found in the appendix under section Switching sequence “All combinations”, page 87.

How does the device know where it currently is in the sequence?

The device determines the current position from the object values.

Next switching step = acual value of the object ± 1

+1 → switch up (increment)

–1 → switch down (decrement)

Is it possible to control a switching sequence in parallel from several push buttons?

Yes, the object “Level increment/decrement” exists for this purpose.

This is a further option which allows you to link two (or more) channels of the same objects with the same group addresses. The channels thus listen in to other communications. It is important that both channels use the same switching sequence.

Example: Switching sequence “Sequentially on/off (bidirectional)” using three communication objects

Switching stage		**Value of the communication objects**		
No.	**Short designation**	**“Value3”**	**“Value2”**	**“Value1”**
0	000	OFF	OFF	OFF
1	001	OFF	OFF	ON
2	011	OFF	ON	ON
3	111	ON	ON	ON
4	011	OFF	ON	ON
5	001	OFF	OFF	ON
0	...			

Short code: ...>000>001>011>111>011>001>...

Function on operation

Only visible in the switching sequence *sequentially on/off (one direction)*. Here you can set whether an operation of the push button switches up or down a level.

In the switching sequence *sequentially on/off (one direction)*, at least two push buttons are required, where one switches up (increments) and the other switches down (decrements).

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

3.2.9.2 Communication objects “Switch sequence”

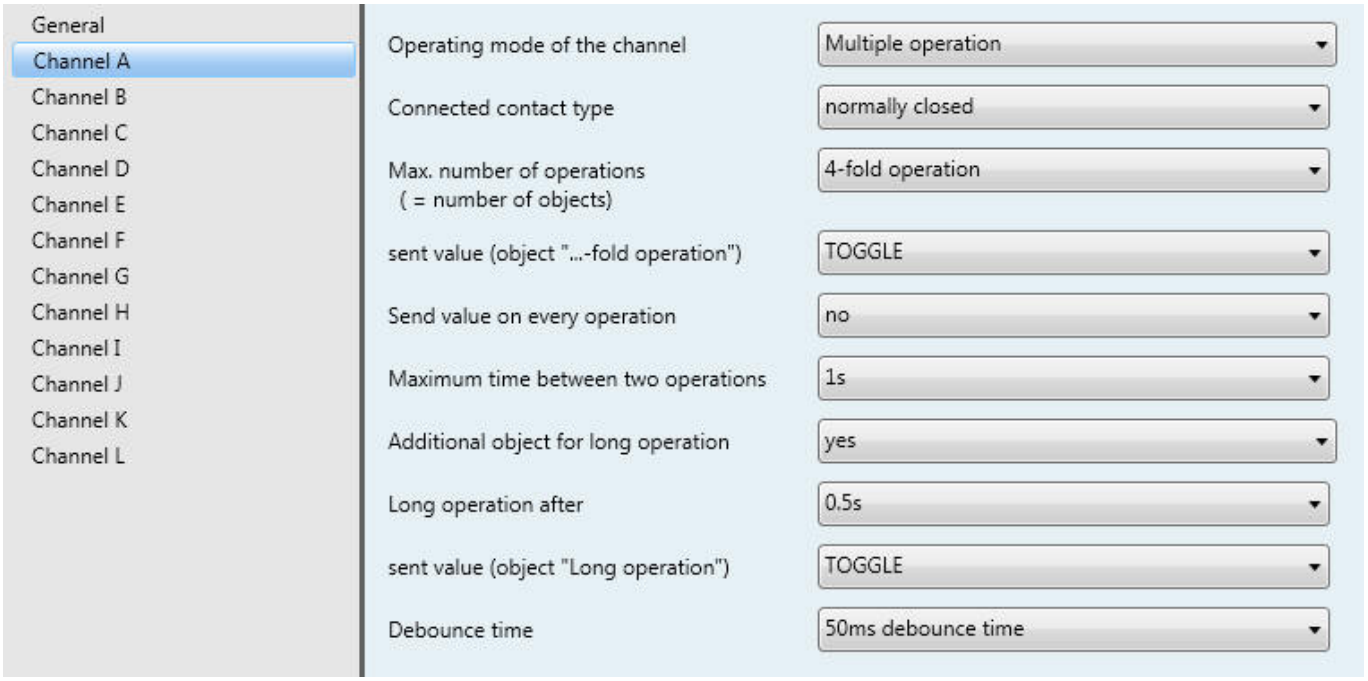
No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 bit** **DPT 1.003**	**C, W**
The function of the input circuitry can be blocked or enabled using the communication object *Block*. A blocked input behaves as if there has been no change of the input signal. The input objects continue to be available. 0: enable input 1: disable input If the input is just being operated as it is blocked, the response is undefined. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the state of the input has changed during blocking. If the input is just being operated as it is being enabled, the input behaves as if the operation has just commenced.				
1 **...** **5**	**Value 1** **...** **Value 5**	**Channel A** **...** **Channel A**	**1 bit EIS1** **DPT 1.001**	**C, T**
The number of these max. 5 objects is set in parameter *Number of values*. The objects represent the values within the switching sequence.				
6	**Level increment/decrement**	**Channel A**	**1 bit** **DPT 1.001**	**C, W**
When an ON telegram is received on this communication object, the input increments a level and when an OFF telegram is received, it decrements a level. 0: Switch down (decrement) 1: Switch up (increment)				

3.2.10 Operation mode “Multiple operation”

The operating mode “Multiple operation” is described in the following.

This operating mode enables detection of multiple operations performed in quick succession and the operation of the switching actions that they trigger.

3.2.10.1 Parameters



Operating mode of the channel	Multiple operation
Connected contact type	normally closed normally open
Max. number of operations (= number of objects)	single operation 2-fold operation 3-fold operation 4-fold operation
sent value (object "...-fold operation")	ON OFF TOGGLE
Send value on every operation	yes no
Maximum time between two operations	0.3 s…1 s…10 s
Additional object for long operation	yes no

Long operation after	0.3 s…0.5 s…10 s
sent value **(object "Long operation")**	ON OFF TOGGLE
Debounce time	10 ms…50 ms…150 ms debounce time

Connected contact type

normally open: Input is closed with actuation (normally open contact).

normally closed: The input is opened with actuation (normally closed contact)

Max. number of operations

The maximum number of posible operations is set here. The number is equal to the number of communication objects *…fold operation*.

Note: If the actual number of operations is greater than the maximum value set here, the input reacts as if the number of operations is equal to the maximum value set here.

sent value

Here the object value to be sent can be set here. The settings *ON*, *OFF* and *TOGGLE* are possible. The current object value is inverted using *TOGGLE*.

Send value on every operation

If *yes* has been entered for this parameter, the respective object value is updated and sent after each operation with multiple operation.

Example: With three-fold operation, the objects *1-fold operation* (after 1st operation), *2-fold operation* (after 2nd operation) and *3-fold operation* (after 3rd operation) are sent.

Maximum time between two operations

The time that can elapse between two operations is set here.

If the device has detected an operation, it will wait for the time entered here. If there are no further operations within this period, counting stops and the object *x-fold operation* is sent. The device then counts again commencing at “1” with the next operation.

Additional object for long operation

A further function can be carried out with long operation of the input via the object *Switch* (long). If a long operation is undertaken within the maximum time after one or more short operations, the short operations are ignored.

Long operation after

In this parameter you set the time period after which an actuation is considered a “long” operation.

sent value

Here you can set with a long operation of the object value *Switch (long)* if “ON“, “OFF“ or “TOGGLE“ is to be switched.

Debounce time

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact. The exact function of this parameter can be found at section 4.1.

3.2.10.2 Communication objects “Multiple operation”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 bit** **DPT 1.003**	**C, W**
The function of the input circuitry can be blocked or enabled using the communication object *Block*. A blocked input behaves as if there has been no change of the input signal. The input objects continue to be available. 0: enable input 1: disable input If the input is just being operated as it is blocked, the response is undefined. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the state of the input has changed during blocking. If the input is just being operated as it is being enabled, the input behaves as if the operation has just commenced.				
1 **...** **4**	**1-fold operation** **...** **4-fold operation**	**Channel A** **...** **Channel A**	**1 bit** **DPT 1.001**	**C, T**
The number of these max. 4 objects is set in parameter “*Max. number of operations”*. After multiple operations of an input the respective object is sent to suit the number of operations. The telegram value can be set in the parameters.				
5	**Long operation**	**Channel A**	**1 bit** **DPT 1.001**	**C, T**
This object is visible if the parameter “*Additional object for long operation*” has been set to the value “*yes*”. After a long operation has been detected the object is sent. The telegram value can be set in the parameters.				

3.3 Programming

The device can be programmed with the Software ETS2 **V1.3** or higher. In order to reduce the programming time of the device by the ETS, it is delivered in a pre-programmed state. During programming, automatic detection determines whether the correct application program is already in the device.

If the device is pre-programmed with another version, which should only be a very rare occurrence, a full download is completed automatically. This may take a few minutes.

4 Special functions

In the following, special functions are explained, whose descriptions were not possible in conjunction with the parameters and objects for reasons of space.

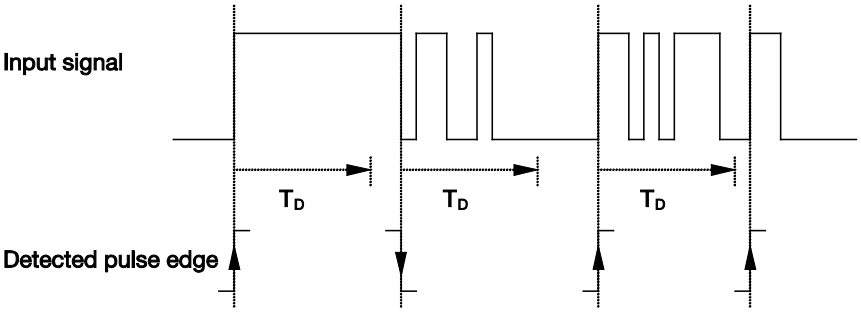
4.1 Debounce time and minimum signal time

A debounce time or a minimum signal time can be defined for each input.

Debounce time

If an edge is detected at an input, the input will react immediately to this edge (e.g. by sending a telegram). The debounce time T_D starts at the same time. When pulse edge are detected at the input during the debounce time they are ignored.

The following example makes this clear:



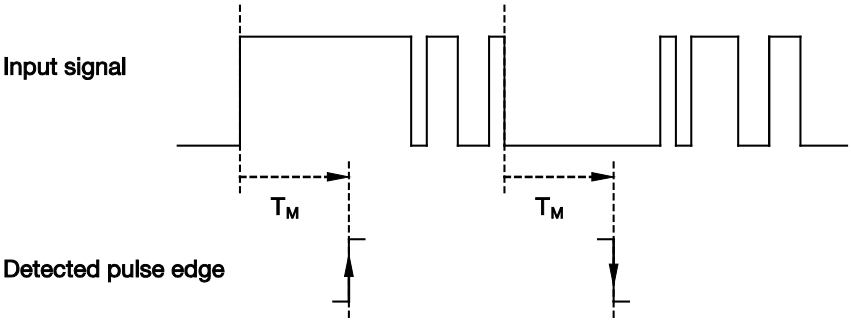
After detection of an edge on the input, further edges are ignored for the debounce time T_D.

Minimum signal time

This function differs from the debounce time in that the telegram is only sent once the minimum operation time has elapsed. The function operates as follows:

If an edge is detected on the input, the minimum signal duration will commence. No telegram is sent on the bus at this time. The signal on the input is observed within the minimum signal duration. If a further edge appears at the input during the minimum signal duration, it will be interpreted as a new operation, and the minimum signal duration restarts if necessary. If the input signal duration has not changed during the minimum signal duration, an edge is detected and a telegram is sent on the bus if required.

The following example makes this clear:



As only two edges remain stable for the minimum signal time T_M, only these are detected as valid.

4.2 Reaction on bus voltage failure

After bus voltage failure, the device switches to energy saving mode for a short period in order to retain the stored values for as long as possible. If the bus voltage recovers during energy saving mode, the status of the device is fully maintained.

After a bus voltage failure of approx. 300 ms (duration is dependent on the function of the device), the energy saving mode is completed and the temporary memory is deleted. All the object values are equal to “0” and the device carries out an initialisation after bus voltage recovery.

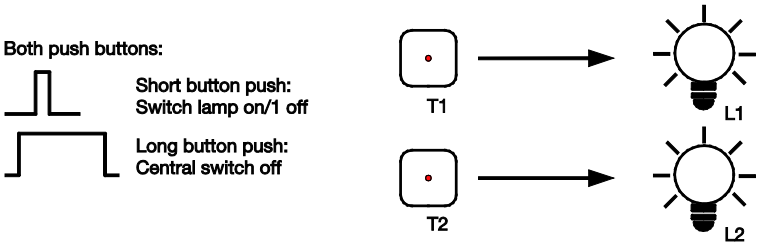
4.3 Reaction after bus voltage recovery

The reaction depends on the operating mode. It can be set in the parameters in most cases. A detailed description can be found in section 3.2.1.1.

5 Application examples

In this section you will find some tips and application examples for practical use of the device.

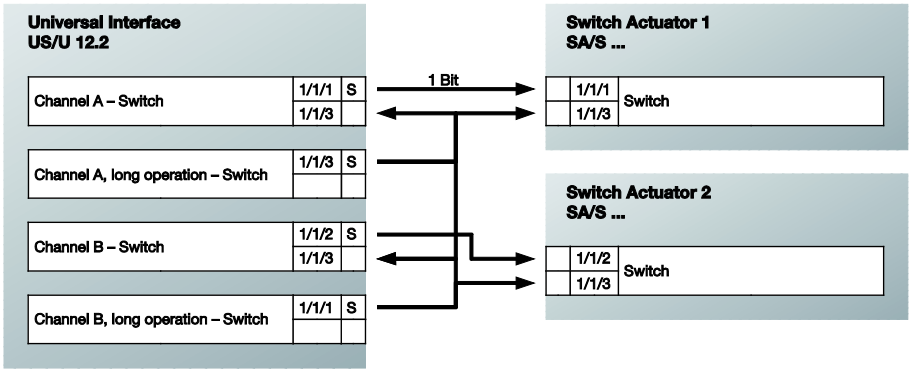
5.1 1 button operation with central function



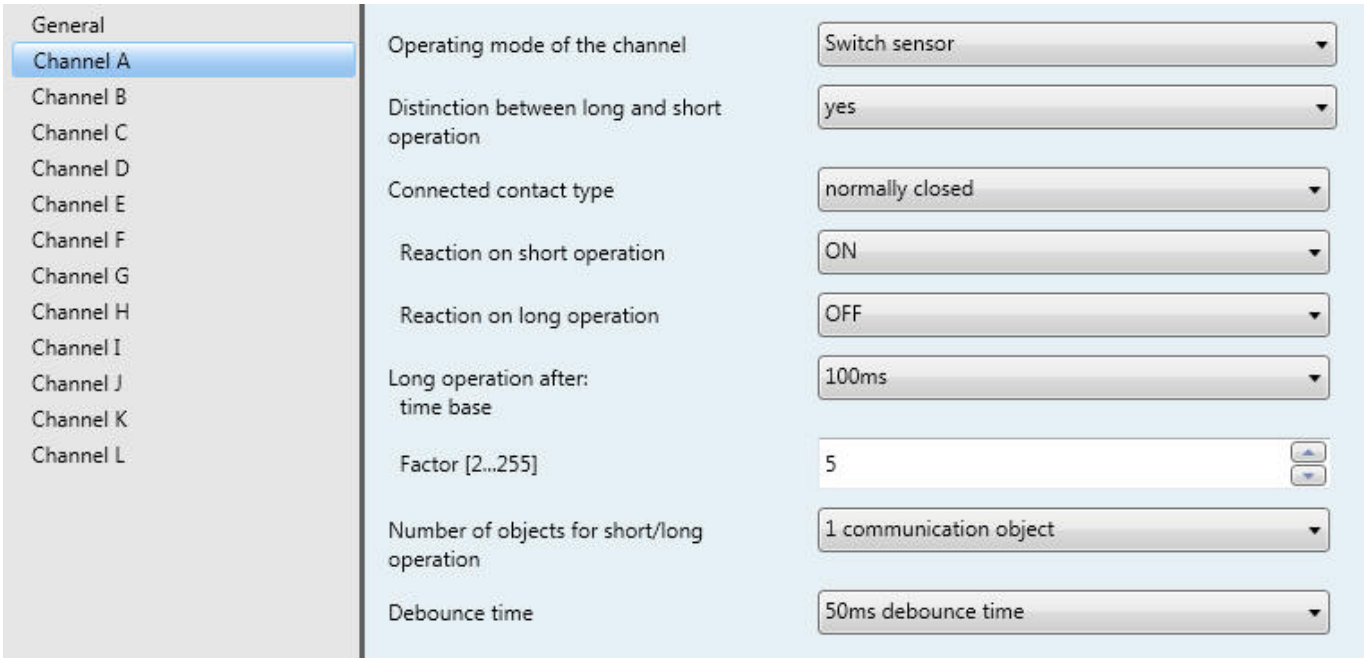
The lighting is switched on and/or off with a short push of the button. A long button push switches both lights off centrally.

T1 is linked with channel A and T2 with channel B.

Logical connection of the group addresses:

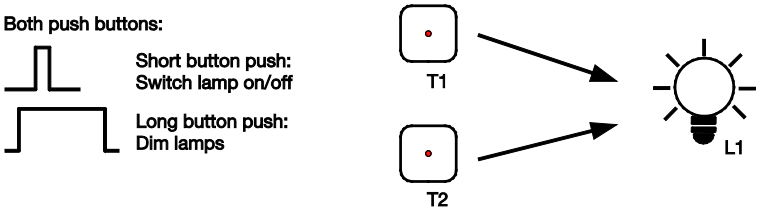


Parameter settings for channel A and B:



5.2 Operation of dimmable illumination

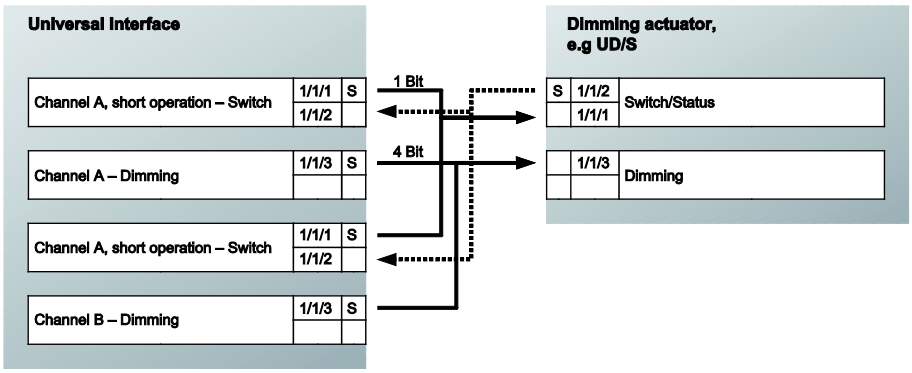
1 button operation



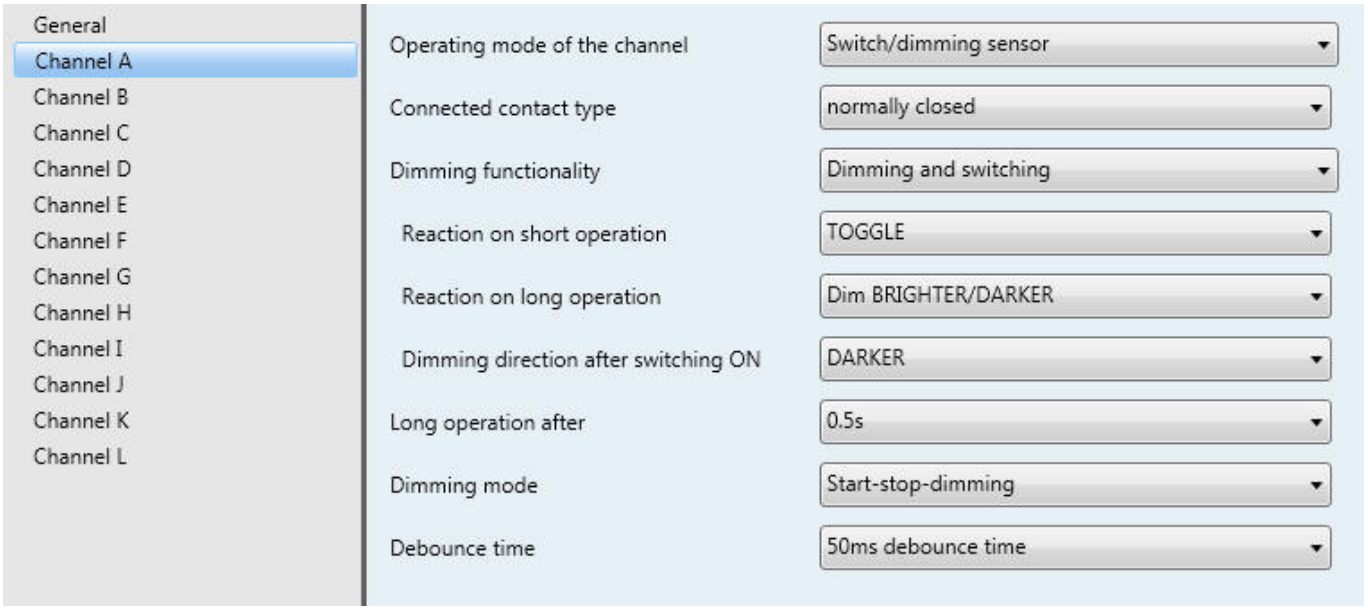
A short button push switches the light centrally. A long button push dims alternately brighter or darker. Both buttons operate the same lamp from different locations.

T1 is linked with channel A and T2 with channel B.

Logical connection of the group addresses (assuming the dimming actuator sends a switch state via the switch object “Switch / Status”):



Parameter settings for channel A and channel B:



2 button operation

The same group address logical connection is also suitable for 2 button dimming: T1 switches on or dims brighter, T2 switches off or dims darker. Only parameters are to be modified:

"*Reaction on short operation*" = "ON" (T1) or "OFF" (T2)

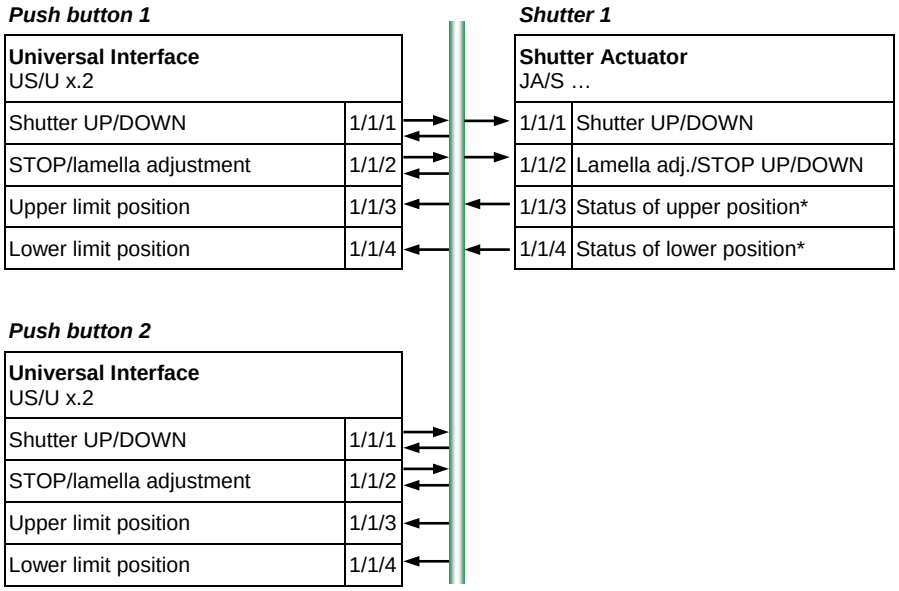
"Reaction on long operation" = "Dim BRIGHTER" (T1) or "Dim DARKER" (T2).

5.3 Operation of shutters

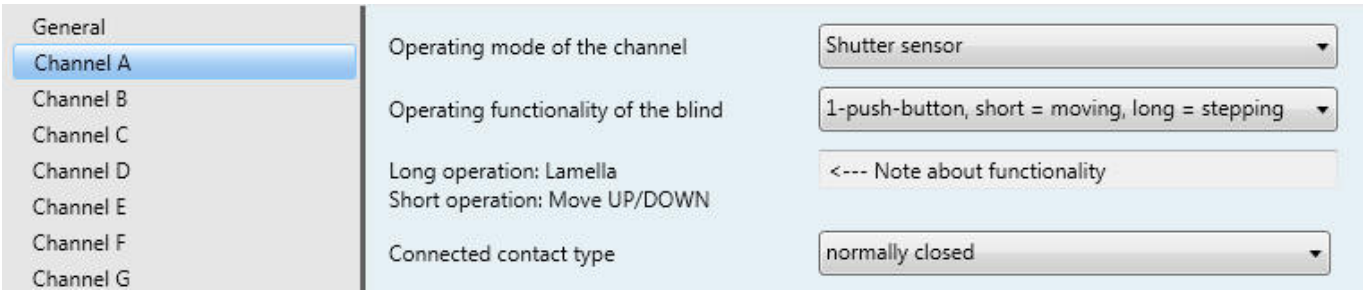
1 button operation

Push button 1 and push button 2 operate shutter 1 from different locations. With a short button operation the shutter moves (in the opposite direction to the last movement); a long operation offsets the louvre.

Logical connection of the group addresses:



Parameter settings for push button 1 and 2:

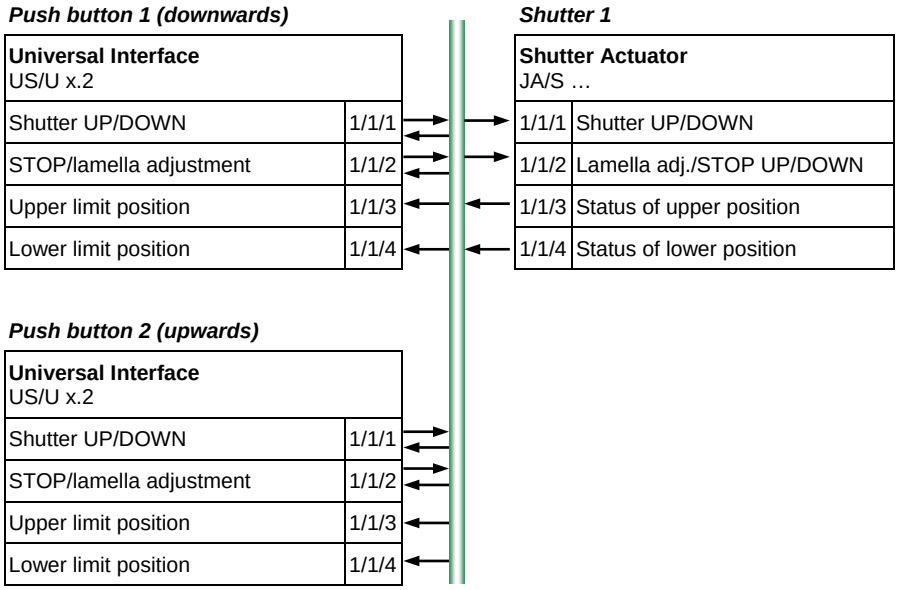


* Via the objects “Upper limit position” and “Lower limit position”, the Universal Interfaces discover if the actuators are in an end limit position. This function is supported from the newer ABB shutter actuator generation (from 2003). Otherwise, 2 button operation is recommended.

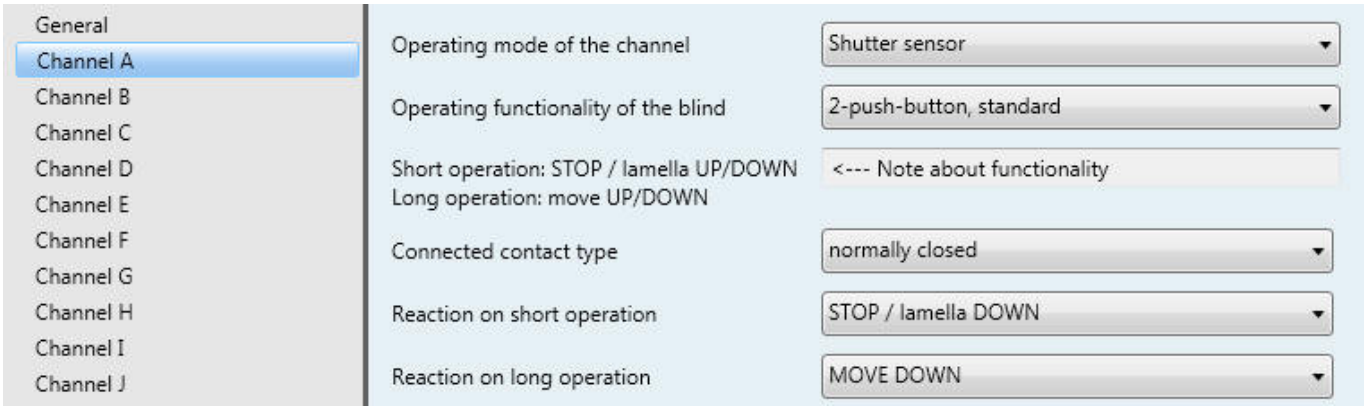
2 button operation

Push button 1 and push button 2 operate shutter 1 from a single location. With long operation the shutter moves DOWN (push button 1) or UP (push button 2). With short operation, the louvre will close (push button 1) or open (push button 2) by a step.

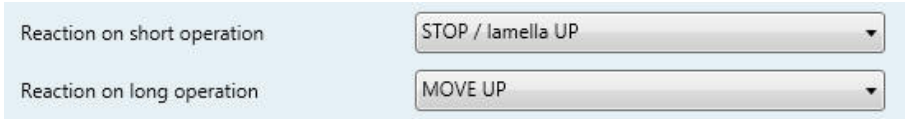
Logical connection of the group addresses:



Parameter settings for push button 1:



Parameter settings for push button 2:

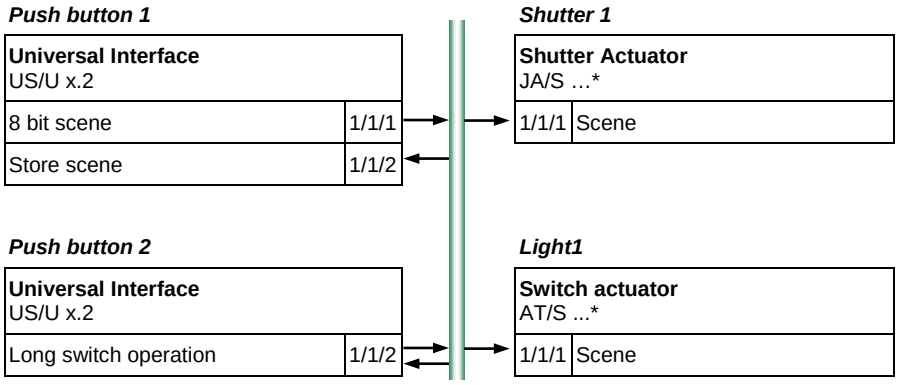


5.4 Control of scenes

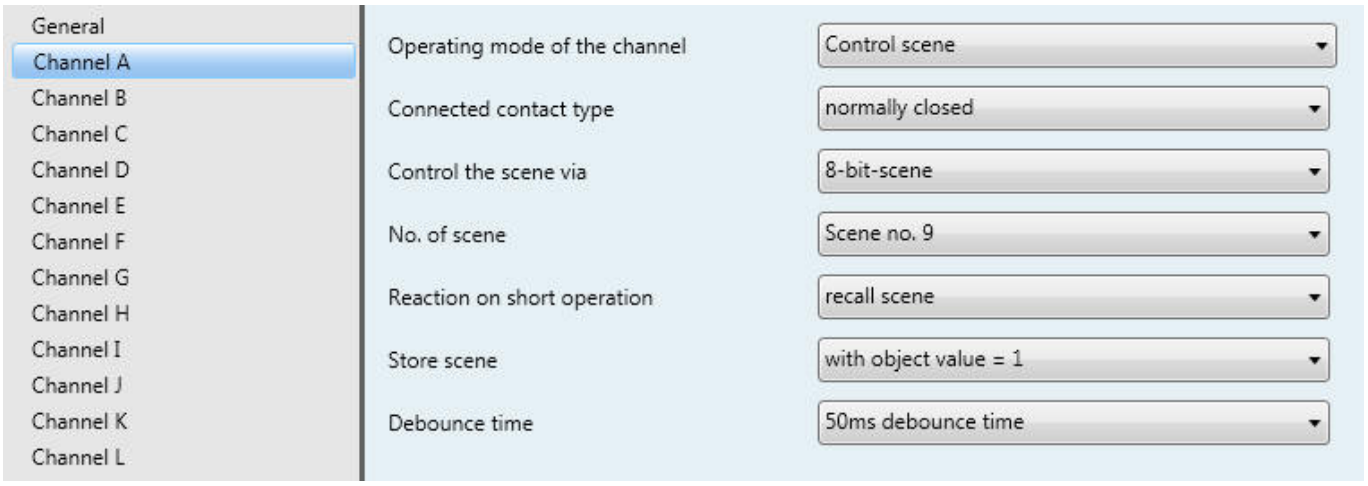
8 bit scene*

Push button 1 and push button 2 control shutter1 and light 1. Push button 1 recalls the scene. On a long operation of push button 2 the cutrrent shutter position and the state of the lighting are stored. The positions are stores in the actuator.

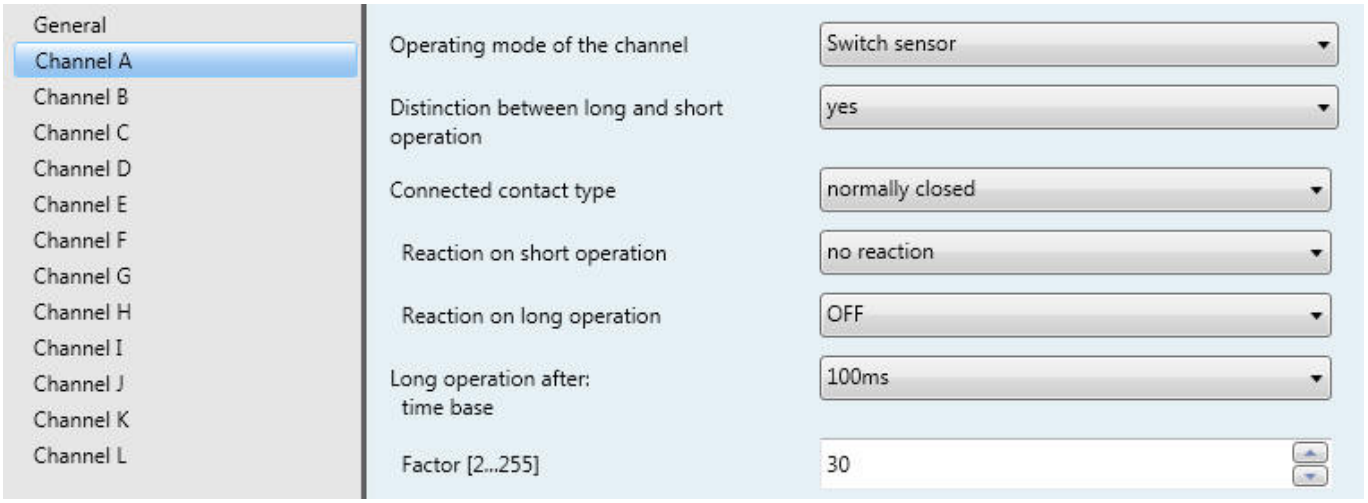
Logical connection of the group addresses:



Parameter settings for push button 1:



Parameter settings for push button 2:

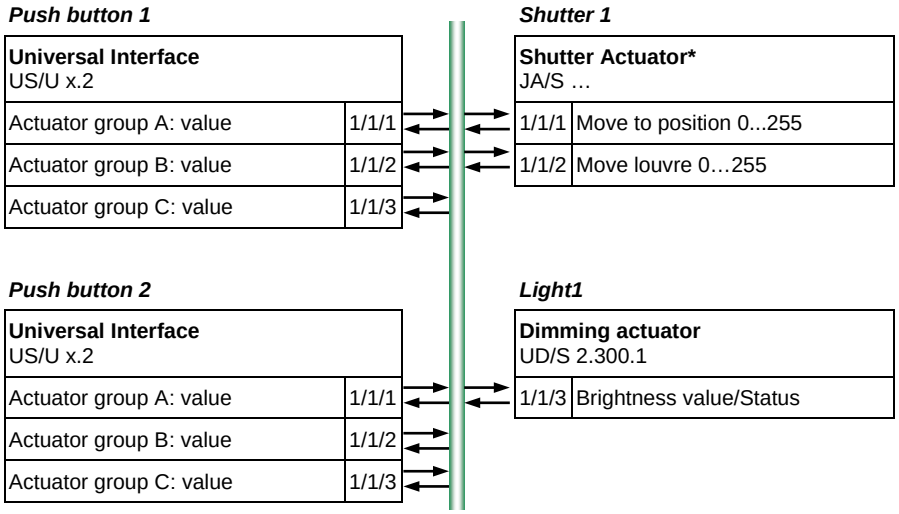


* The 8 bit scene requires actuators that support these functions. This is the case for ABB shutter actuators and switch actuators of the latest generation (from 2003). On other devices the scene is recommended via “5 separate objects”.

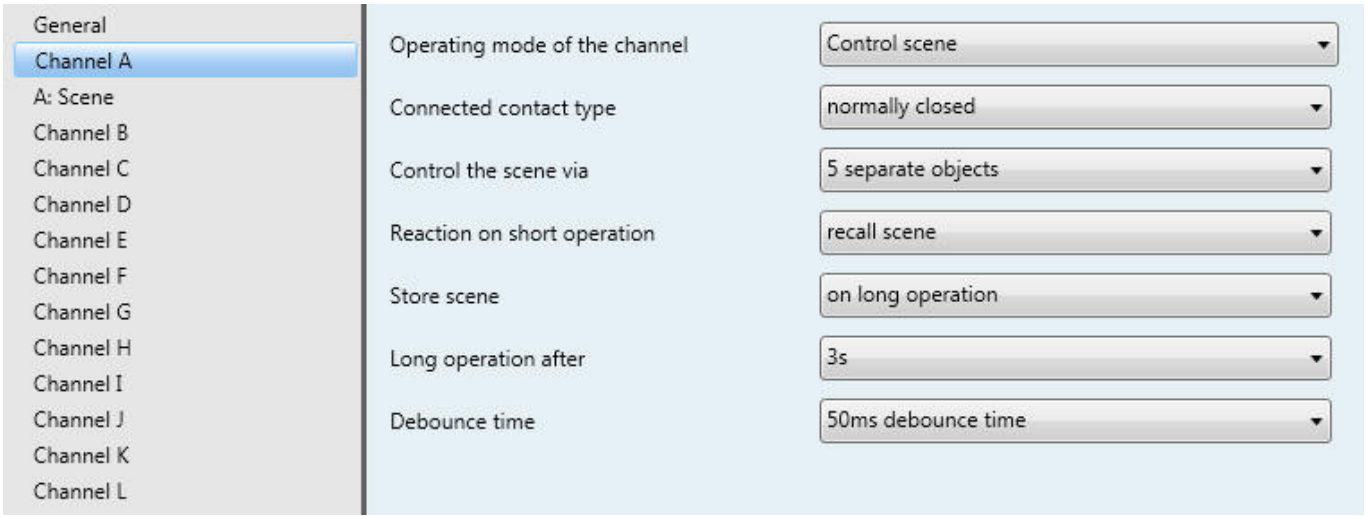
Scene via 5 separate objects

Push button 1 and push button 2 control Shutter 1 and Light 1. Short operation recalls the scene. On long operation the current shutter setting and brightnes value are stored. Both push buttons store different scene values.

Logical connection of the group addresses:



Parameter settings for push button 1 and 2:

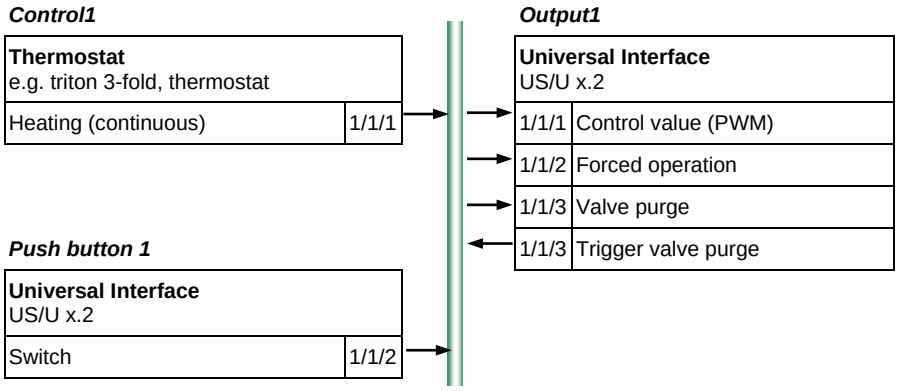


* This function is only available for shutter actuators, which can move to a position via an 8 bit value.

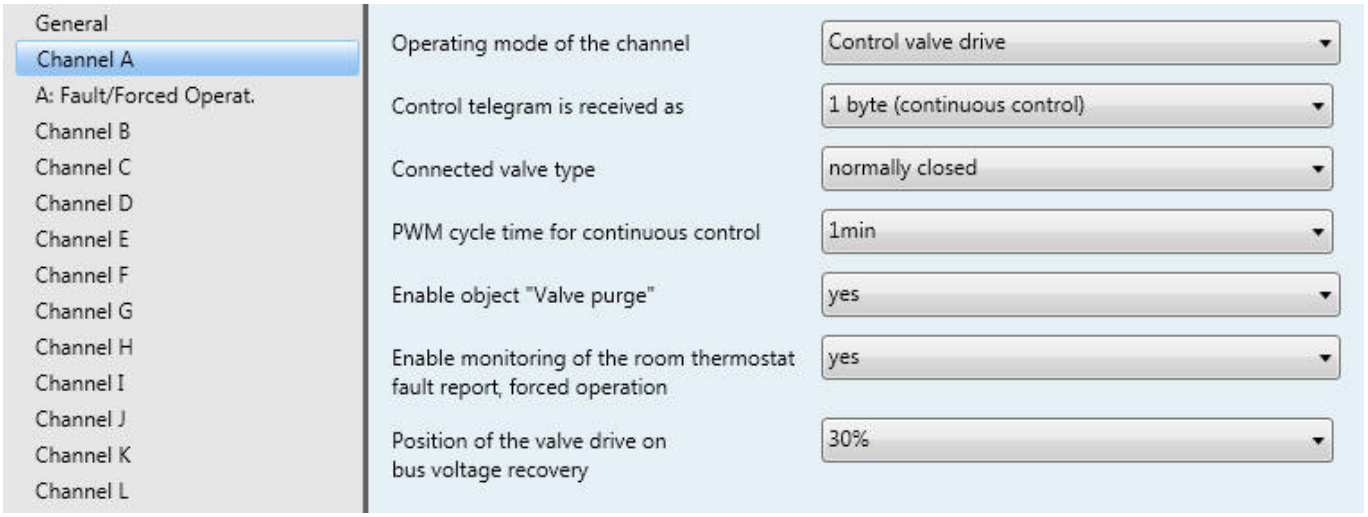
5.5 Control of a heater valve

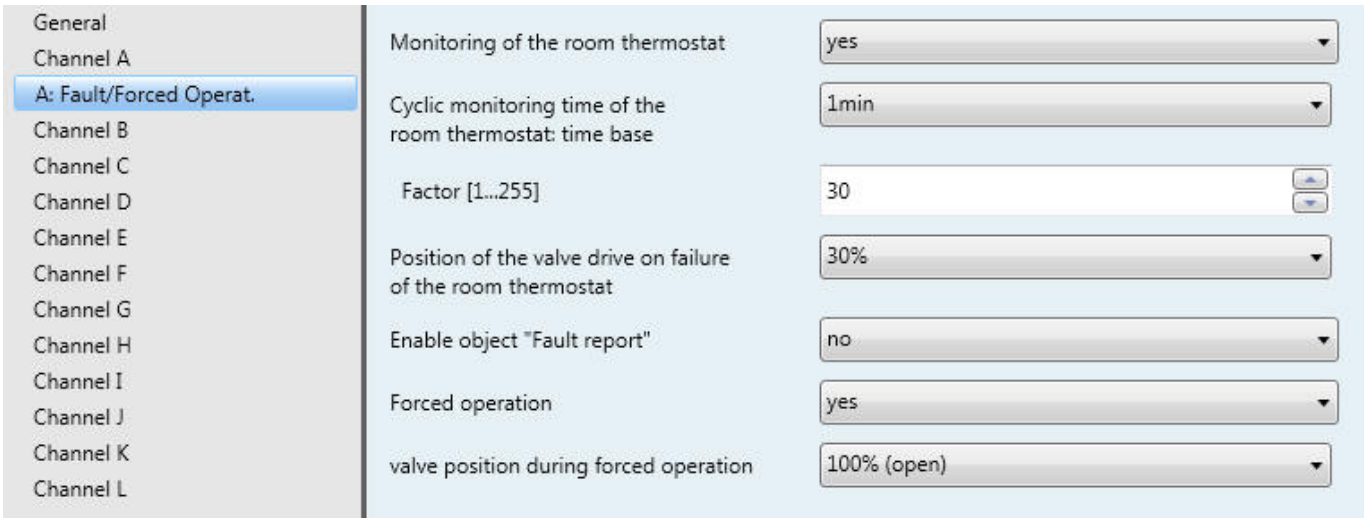
An Electronic Relay ER/U 1.1 is connected to output 1 of a Universal Interface, which controls an electro-thermal valve drive. The room temperature is continuously controlled via control 1. Once a week the valve is purged by opening it for about 5 minutes. The valve can be forcibly fully opened via push button 1. If no telegram has been received from control 1 for 30 minutes, the valve sets to 30% open (fault operation).

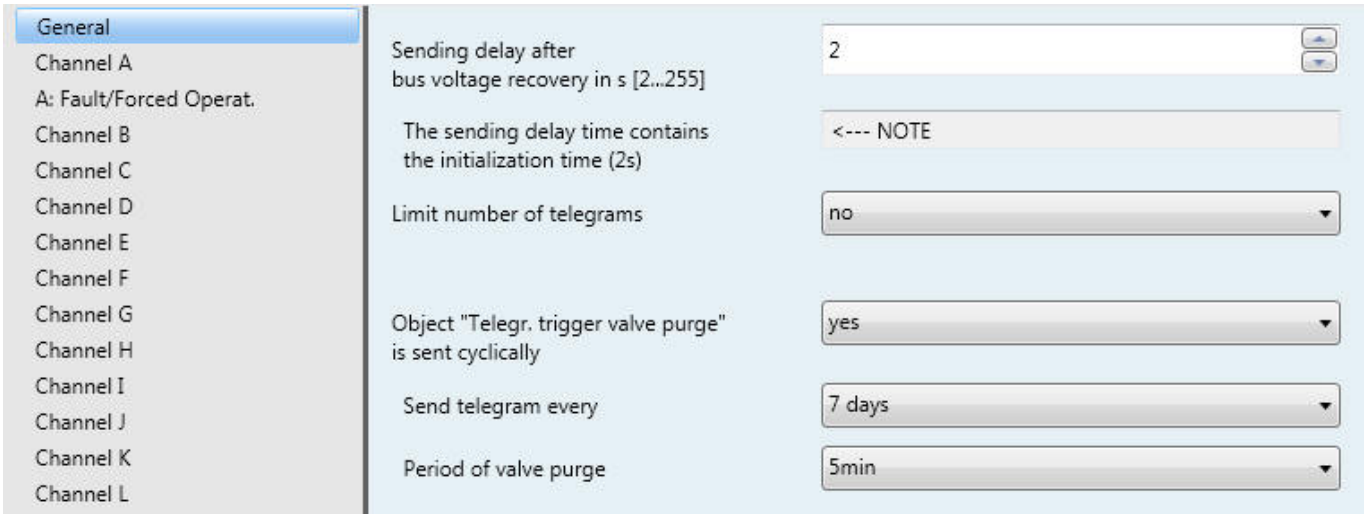
Logical connection of the group addresses:



Parameter settings for output 1:







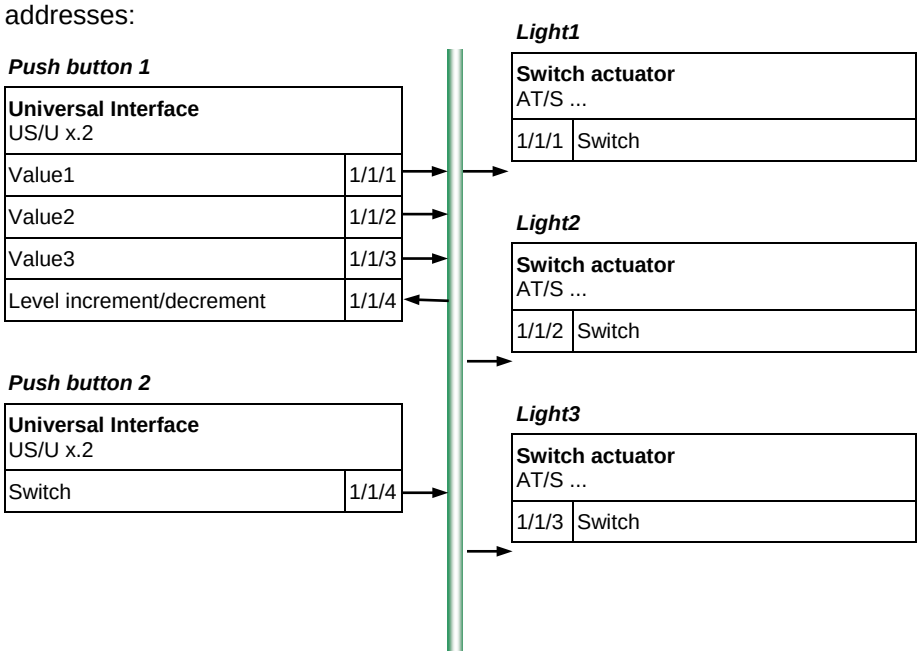
Output 1 sends itself the group address 1/1/3 once a week and thus triggers valve purging. The sending object is enabled in parameter window “General”.

5.6 Switching of lighting in switching sequence

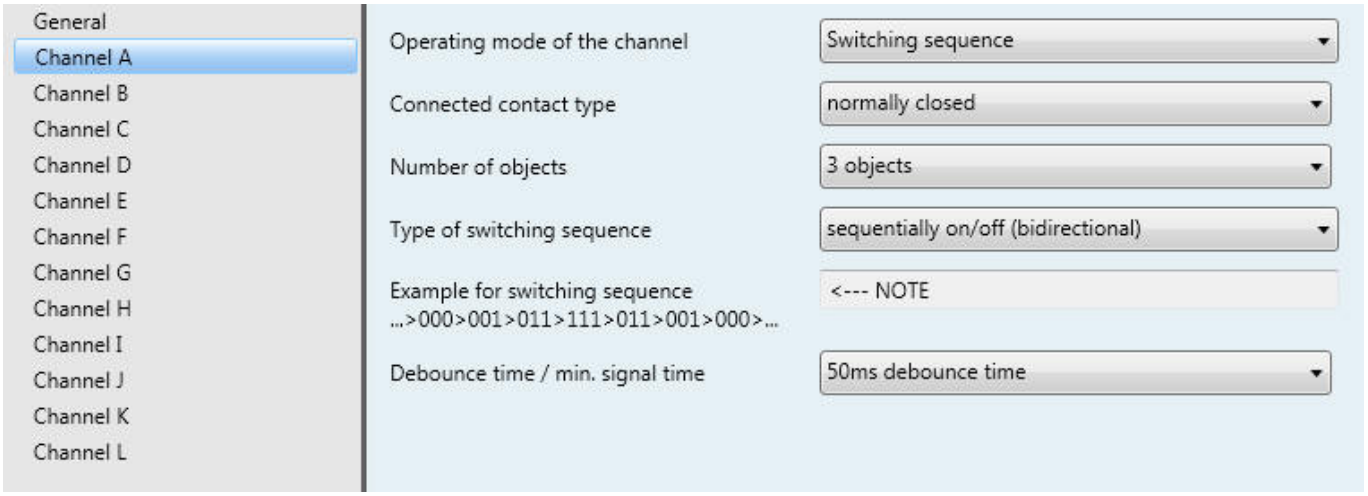
Successive switch on/off

Push button 1 and push button 2 control a lamp with three independent circuits Light 1, Light 2 and Light 3. Push button 1 switches on the actuation successively (sequence: Light1>Light2>Light3). Push button 2 switches off the actuation successively (sequence: Light3>Light2>Light1).

Logical connection of the group addresses:



Parameter settings for push button 1:



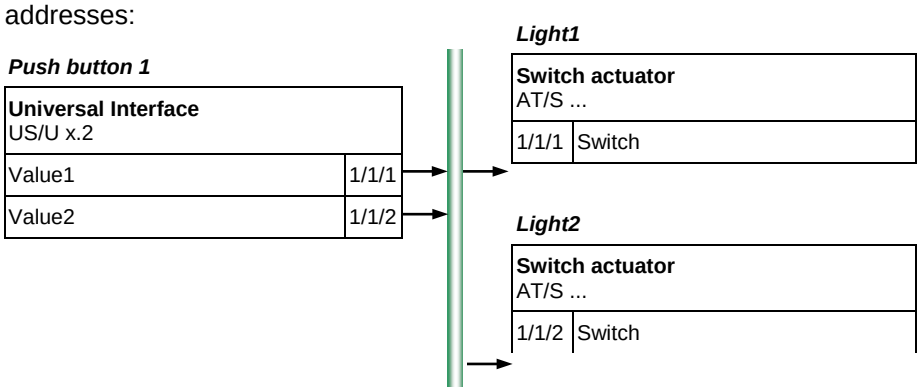
Push button 2 has to be parameterized so that “Switch” sends a “0” with very button push.

Switch all combinations

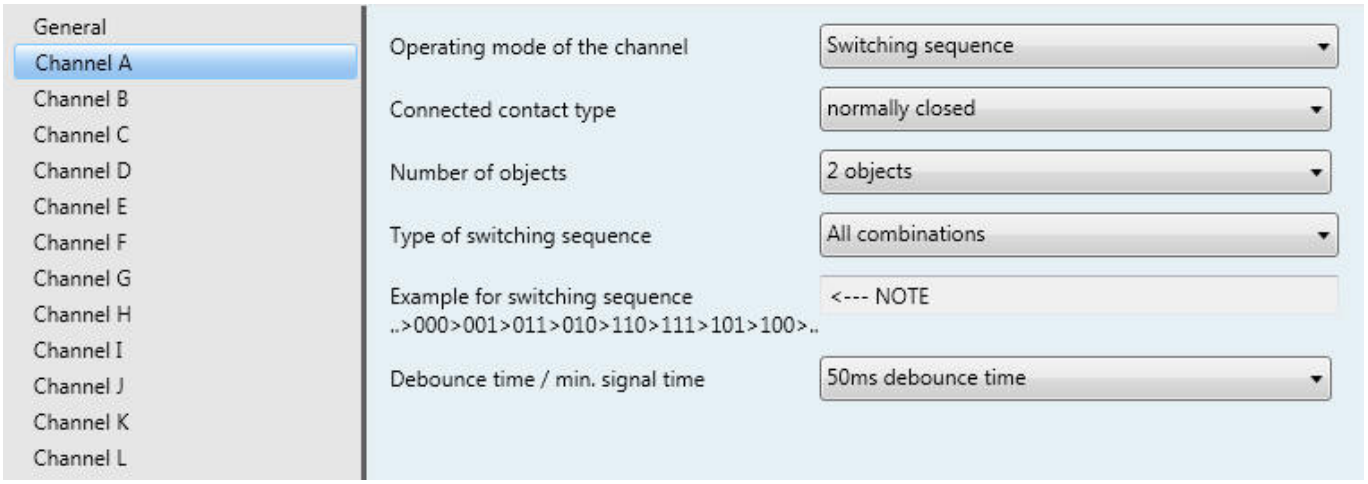
Push button 1 controls a lamp with two independent circuits Light 1 and Light 2. On actuation, all combinations in the following sequence are switched through:

	Light1	*Light2*
Initial state	OFF	OFF
1st operation	ON	OFF
2nd operation	ON	ON
3rd operation	OFF	ON
4th operation	OFF	OFF
... (and so forth)		

Logical connection of the group addresses:



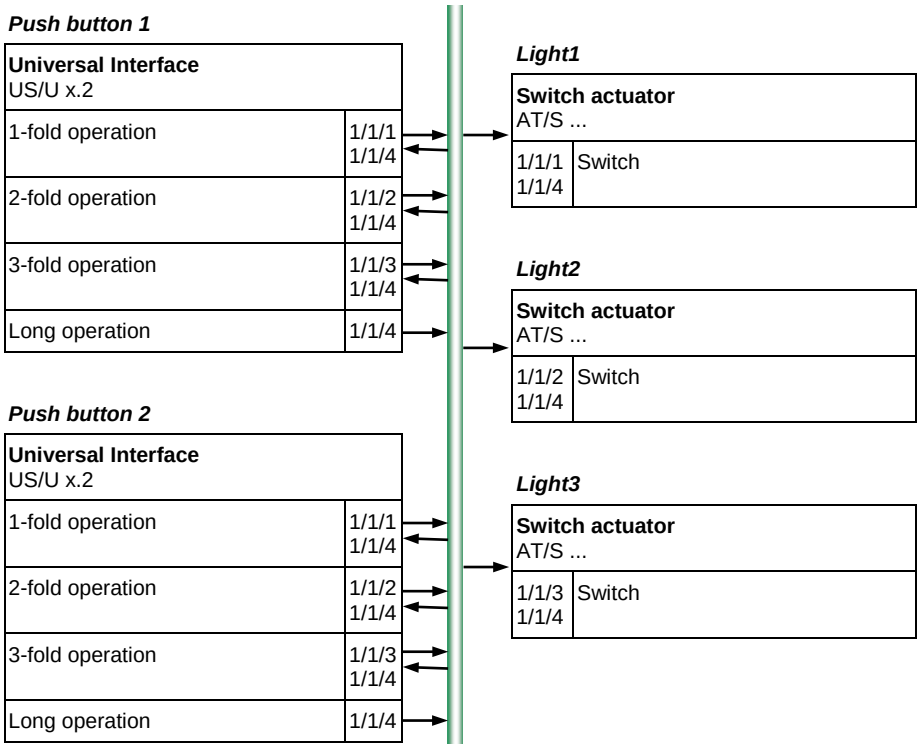
Parameter settings for push button 1:



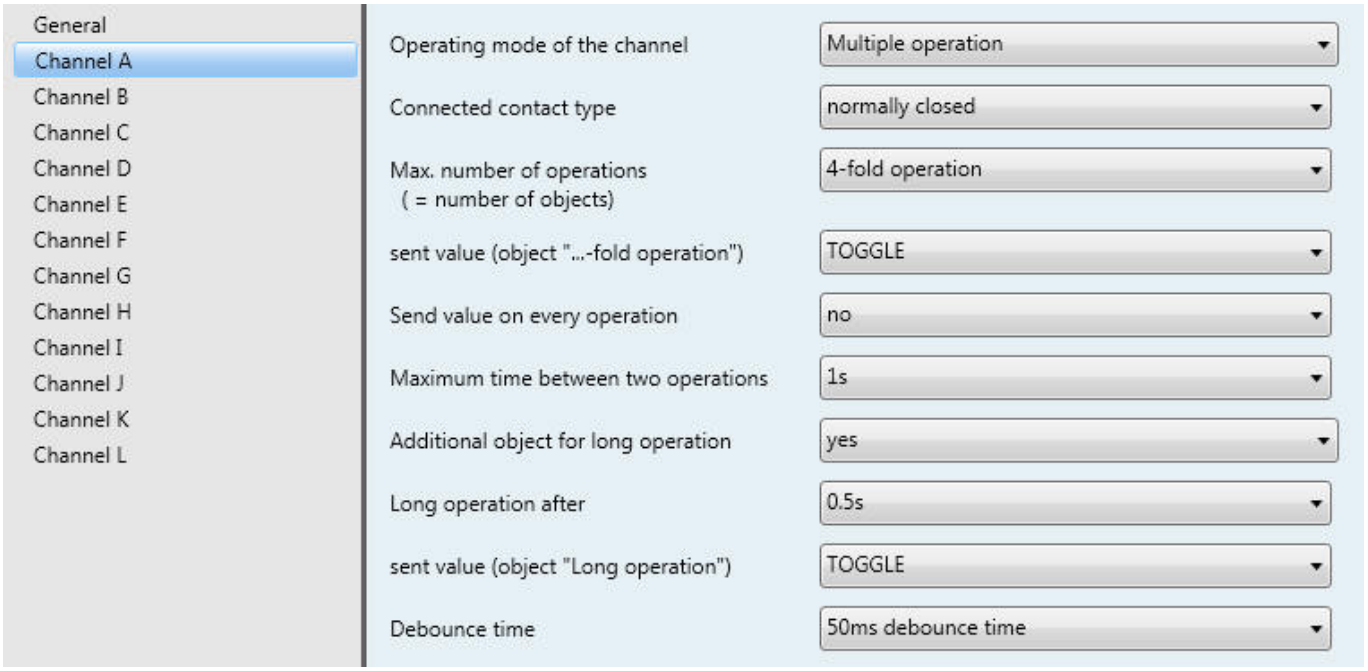
5.7 Switching of lighting via multiple button pushes

Push button 1 and push button 2 control Light1, Light2 and Light3. With a single button push Light1 is switched, with a 2-fold button push Light2 is switched and with a 3-fold button push Light3 is switched. With a long button push, Light1, Light2 and Light3 are switched off.

Logical connection of the group addresses:



Parameter settings for push button 1 and 2:



6 Appendix

6.1 Switching sequence “All combinations”

The switching sequence “All combinations” undertakes all options in succession. Only one value changes between two stages and thus only one telegram is sent.

The following table describes the sequence when 5 objects are used:

Switching stage		**Value of the communication objects**				
No.	**Short designation**	**Value 5**	**Value 4**	**Value 3**	**Value 2**	**Value 1**
0	00000	OFF	OFF	OFF	OFF	OFF
1	00001	OFF	OFF	OFF	OFF	ON
2	00011	OFF	OFF	OFF	ON	ON
3	00010	OFF	OFF	OFF	ON	OFF
4	00110	OFF	OFF	ON	ON	OFF
5	00111	OFF	OFF	ON	ON	ON
6	00101	OFF	OFF	ON	OFF	ON
7	00100	OFF	OFF	ON	OFF	OFF
8	01100	OFF	ON	ON	OFF	OFF
9	01101	OFF	ON	ON	OFF	ON
10	01111	OFF	ON	ON	ON	ON
11	01110	OFF	ON	ON	ON	OFF
12	01010	OFF	ON	OFF	ON	OFF
13	01011	OFF	ON	OFF	ON	ON
14	01001	OFF	ON	OFF	OFF	ON
15	01000	OFF	ON	OFF	OFF	OFF
16	11000	ON	ON	OFF	OFF	OFF
17	11001	ON	ON	OFF	OFF	ON
18	11011	ON	ON	OFF	ON	ON
19	11010	ON	ON	OFF	ON	OFF
20	11110	ON	ON	ON	ON	OFF
21	11111	ON	ON	ON	ON	ON
22	11101	ON	ON	ON	OFF	ON
23	11100	ON	ON	ON	OFF	OFF
24	10100	ON	OFF	ON	OFF	OFF
25	10101	ON	OFF	ON	OFF	ON
26	10111	ON	OFF	ON	ON	ON
27	10110	ON	OFF	ON	ON	OFF
28	10010	ON	OFF	OFF	ON	OFF
29	10011	ON	OFF	OFF	ON	ON
30	10001	ON	OFF	OFF	OFF	ON
31	10000	ON	OFF	OFF	OFF	OFF

6.2 Value table for object "8 Bit scene"

Object value		Meaning
Decimal	**Hexadecimal**	
00 *or* 64	00h *or* 40h	Recall scene 1
01 *or* 65	01h *or* 41h	Recall scene 2
02 *or* 66	02h *or* 42h	Recall scene 3
...	...	...
63 *or* 127	3Fh *or* 7Fh	Recall scene 64
128 *or* 192	80h *or* B0h	Store scene 1
129 *or* 193	81h *or* B1h	Store scene 2
130 *or* 194	82h *or* B2h	Store scene 3
...	...	...
191 *or* 255	AFh *or* FFh	Store scene 64

6.3 Ordering information

Designation	**Ordering information** **Short description**	**Order No.**	**bbn 40 16779 EAN**	**Price 1 pc. [EURO]**	**Price group**	**Weight 1 pc. [kg]**	**Pack unit [pc.]**
Universal Interface, 12-fold	US/U 12.2	2CDG 110 065 R0011			26	0.05	1

Contact

ABB STOTZ-KONTAKT GmbH
Eppelheimer Straße 82
69123 Heidelberg, Germany
Phone: +49 (0)6221 701 607 (Marketing)
Fax: +49 (0)6221 701 724
e-mail: knx.marketing@de.abb.com

Further information and local contacts:
www.abb.com/knx

Note:
We reserve the right to make technical changes to the products as well as amendments to the content of this document at any time without advance notice.
The agreed properties are definitive for any orders placed. ABB AG shall not be liable for any consequences arising from errors or incomplete information in this document.

Publication Number 2CDC 504 079 D0201 (08.12)

Power and productivity
for a better world™