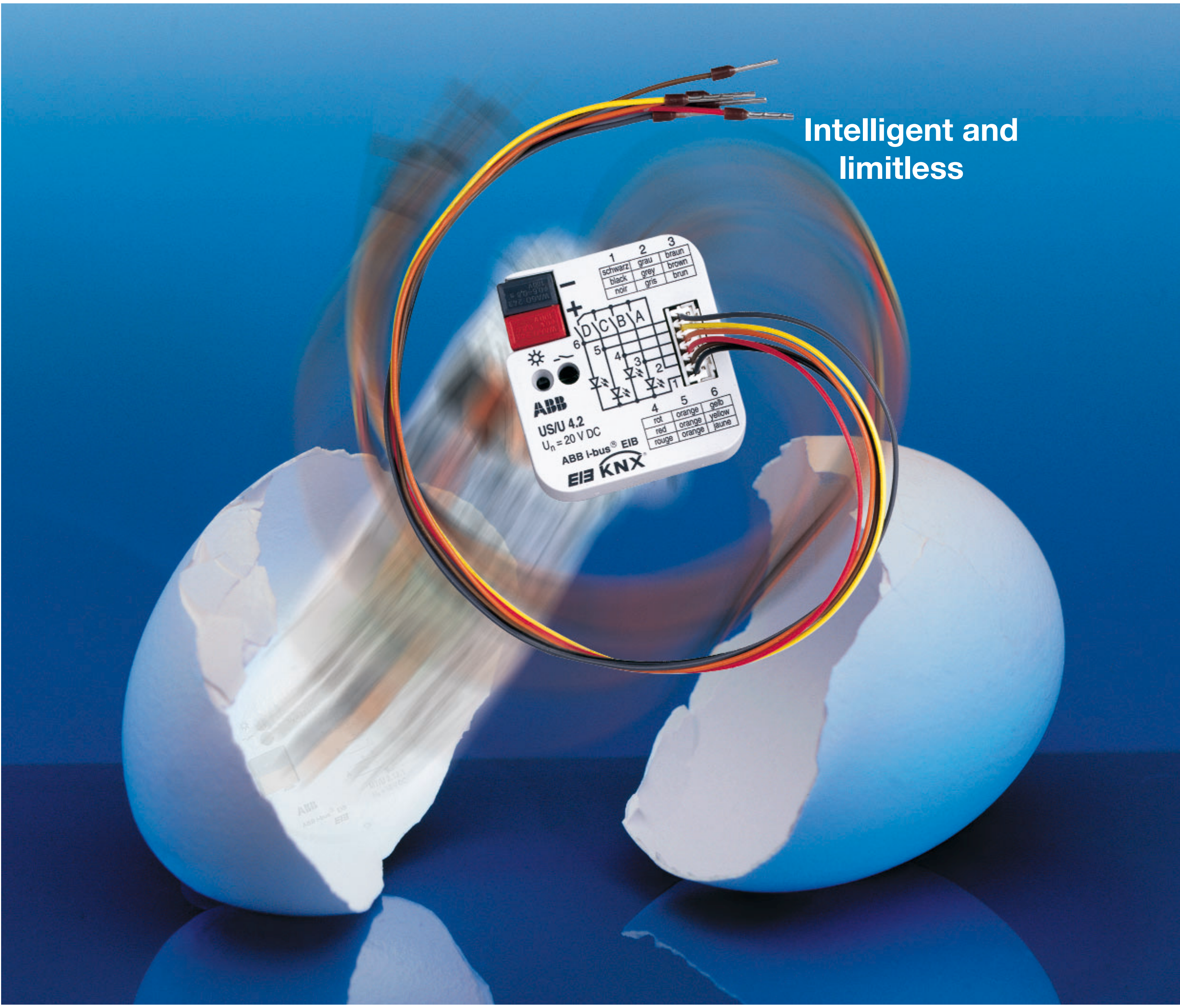
Product Manual

ABB i-bus® EIB

Universal Interfaces
US/U 4.2
US/U 2.2

Intelligent Installation Systems



ABB

Contents

Contents

This manual describes the function of the universal interfaces US/U 2.2 and US/U 4.2 with the application program “Binary Input Display Heat xf/1”.

Subject to technical changes. Errors excepted

Exclusion of liability:

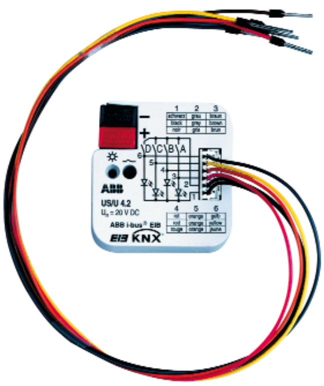
Despite checking that the contents of this document correspond to the hardware and software, deviations cannot be ruled out completely. We therefore cannot accept liability for this. Any necessary corrections will be integrated in new versions of the manual.

Please tell us about any suggestions for improvement you may have.

ABB i-bus® EIB

Universal Interfaces
US/U 2.2, GH Q631 0074 R0111
US/U 4.2, GH Q631 0070 R0111

1 General



US/U 4.2

SK 0093 B02

The functions implemented in modern buildings with ABB i-bus® EIB should be both simple to operate and intuitive. At the same time, clear and user-friendly operation is extremely important to the value of a building installation.

The universal interfaces US/U 2.2 and US/U 4.2 meet individual requirements in both functional buildings and the residential sector. A variety of application possibilities are thus available to the planner of the system as regards the implementation of functions.

This manual provides technical information about the device as well as its assembly and programming. The last section contains application examples for its effective use on site.

1.1 Product and functional overview

The universal interfaces US/U 2.2 (two channels) and US/U 4.2 (four channels) are used as an interface for the convenient operation of ABB i-bus® EIB installations via conventional push buttons/switches or for reading out technical binary signals. They also enable the control of LEDs as well as the electronic relay ER/U 1.1 for controlling electrothermal valve drives.

The extremely compact design enables the device to be inserted in a conventional 60 mm wiring box e.g. behind a conventional push button or switch.

The devices stand out due to their functionality which is extremely extensive but comprehensible and enables the devices to be used in a wide variety of application areas. The following list provides an overview:

- Switching and dimming of the lighting (also 1 button operation)
- Operation of blinds and shutters (also 1 button operation)
- Sending of values e.g. temperature values
- Control and storing of lightscenes
- Triggering an electronic relay for controlling an electrothermal drive mechanism for heating valves
- Triggering an LED (with flashing function and time restriction) for reporting an operation
- Operation of various loads by multiple push button actions
- Operation of several loads in a fixed switching sequence
- Counting of impulses and push button operations
- Reading out of technical contacts (e.g. relays)

Each channel of a device can adopt any of the functions described above.

2 Device technology

This section outlines the device functions of the universal interfaces US/U 2.2 and US/U 4.2. The device has two (US/U 2.2) or four (US/U 4.2) channels which can be configured as inputs or outputs with the EIB Tool Software **ETS2 V1.2a** (or higher).

Using the colour-coded connecting cables, it is possible to connect conventional push buttons, floating contacts, light-emitting diodes (LEDs) or electronic relays ER/U 1.1. Series resistors for the operation of LEDs are integrated into the device. The contact scanning voltage and the supply voltage for LEDs or electronic relay are made available by the device.

The bus connection is carried out via the bus connecting terminal supplied.

2.1 Technical data

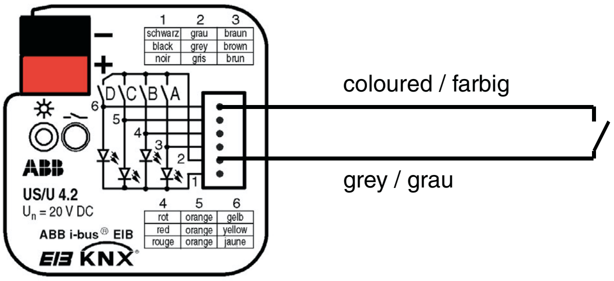
Power supply:	– Bus voltage	via the ABB i-bus® EIB
		Power consumption < approx. 10 mA
Inputs/outputs:	– Number	2 for US/U 2.2
		4 for US/U 4.2
		Can be individually parameterised
		as inputs or outputs
	– Permitted cable length	≤ 10 m
Input:	– Scanning voltage	20 V DC
	– Input current	0.5 mA
Output:	– Output voltage	5 V DC
	– Output current	max. 2 mA, limited via series resistor
		of 1.5 kΩ
	– Safety	Short-circuit-proof, overload
		protection, reverse voltage protection
Operating and display elements:	– (Red) LED and push button	For assigning the physical address
Connections:	– Inputs/outputs	4 cables for US/U 2.2
		6 cables for US/U 4.2
		approx. 30 cm long, can be extended to max. 10 m
	– ABB i-bus® EIB	via bus connecting terminal,
		with supply
Ambient temperature range:	– Operation	– 5 °C ... 45 °C
	– Storage	– 25 °C ... 55 °C
	– Transport	– 25 °C ... 70 °C
Miscellaneous:	Type of protection	IP 20 (EN 60529)
		when installed
	Protection class	III
	CE norm	in accordance with the EMC
		guideline and the low voltage guideline
	Certification	EIB-certified
	Mounting	in switch box, 60 mm
	Dimensions (W x H x D)	39 x 40 x 12 mm
	Weight	0.05 kg

2.2 Device connection

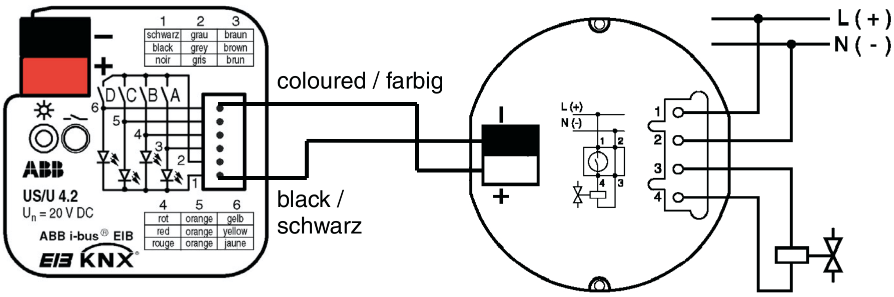
During operation as an input, the scanned contact is connected between the grey core and the coloured core.

During operation as an output, the load (LED or electronic relay) is connected between the black and the coloured core. The coloured core represents the positive output voltage.

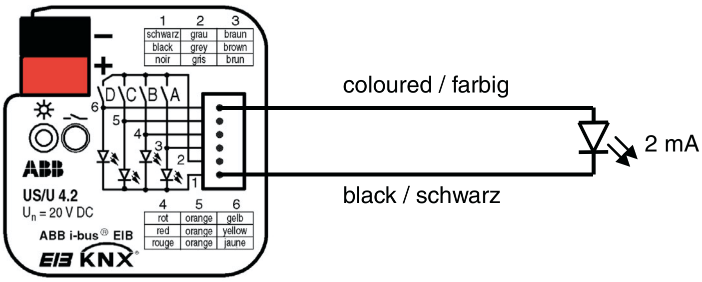
Connection of a floating push button/switch



Connection of an electronic relay



Connection of an LED



Note: When connecting to an S0 impulse output (e.g. an energy consumption meter), it must be ensured that it is electrically isolated from the mains. The correct polarity should also be observed (“+” on the grey core).

2.3 Description of the inputs and outputs

Grey core: Positive scanning voltage

During operation as an input, the grey core makes the positive, pulsed scanning voltage available.

Coloured core: Control of the channel

During operation as an input, the status of the contact is read out via the coloured cores.

During operation as an output, the coloured core makes the positive output voltage available.

The following table allocates the colours to the channels:

brown	Channel A
red	Channel B
orange*	Channel C
yellow*	Channel D

*only for US/U 4.2

Black core: Negative reference potential

During operation as an output, the black core makes the negative reference potential available.

Important: The inputs and outputs do not have electrical isolation to the EIB bus voltage (SELV). The SELV criteria only enable the connection of floating contacts with safety separation.

2.4 Assembly and installation

The device can in mounted in any position. Any cores that are not required must be insulated.

3 Function and operation

This section does not apply.

4 Project design and programming

4.1 Overview of the functions

A powerful application program *"Binary Input Display Heat 2f/1"* (US/U 2.2) and *"… 4f/1"* (US/U 4.2) is available for the universal interfaces. The programming requires ETS2 **V1.2a** or higher.

Application program	Number of communication objects	Max. number of group addresses	Max. number of associations
Binary Input Display Heat, 2f/1	15	254	254
Binary Input Display Heat, 4f/1	29	254	254

The following functions can be set separately for each input:

Switch sensor	For switching the lighting or scanning 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 shutter or blind. Eight present operation modes are possible in total.
Value / Forced operation	For sending the values of different data types (e.g. temperature values). It is possible to send different values or data types for short/long operation, possible to activate/deactivate the forced operation of actuators.
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 of electronic relay (heating actuator)	For controlling an electrothermal valve drive via an electronic relay ER/U 1.1. The device has the full functionality of a heating actuator. Control via 2-step 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 a light-emitting diode. Switching and flashing (with time limit and various flashing rates) and use as an orientation light are possible.
Switching sequence ("latching relay")	For the operation of several actuator groups in a preselected sequence.
Push button with multiple operation	For triggering various functions depending on the frequency of the operation. A long operation can also be detected and a function can be triggered.
Counter	For counting input pulses. Various data types of the counter can be set. An additional intermediate counter enables the counting of e.g. daily values. Factors/dividers enable various counting rates.

4.2 Overview of the communication objects

Both the communication objects and the objects are identical for each channel.

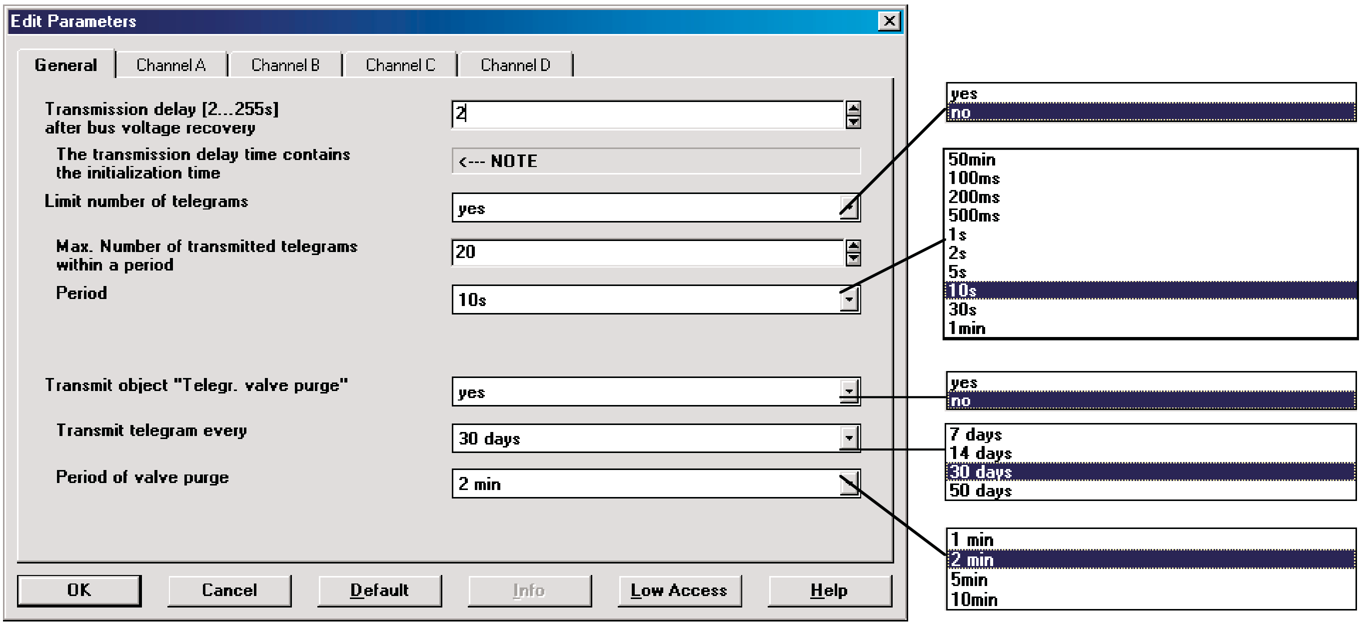
4.3 General functions

Parameters and objects, which apply to the device as a whole, are outlined in this section.

Parameters and objects which are assigned to each channel, are described in the following sections using output A as an example.

4.3.1 General parameters

Parameters for the functions which affect the complete device can be set via the *"General"* parameter window.



Parameter: "Transmission delay after bus voltage recovery"

The transmission delay 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 transmission delay.

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

See section 5.10 for a detailed description of the behaviour on bus voltage recovery.

Parameter: " 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 can be sent within an adjustable period
("Max. number of transmitted telegrams within a period").

Detailed information about the telegram limit function can be obtained under section 5.2.

Parameter: "Transmit object 'Telegr. valve purge'"

This function is only relevant if one or several channels are used to control an electronic relay. Regular purging of a heating control valve can prevent deposits from building up in the valve thereby restricting the valve function. This is particularly significant during periods when only a few changes are made to the valve position.

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

4.3.2 General communication objects

Object "Disable": 1 bit (EIS 1)

This object is visible for each channel that is operated as an input.

The function of the protective input circuit can be disabled or enabled via the "Disable" object. A disabled input behaves as if a change in the input signal has not taken place. The objects of the input remain available.

If the input is disabled during operation, the behaviour is undefined.

When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the input is operated when it is enabled, it behaves as if the operation had started with the end of the blocking.

Telegram value	"0":	Enable input
	"1":	Disable input

Object "Telegr. trigger valve purge": 1 bit (EIS 1)

This object is visible if the parameter ***"Transmit object 'Telegr. valve purge'"*** is set to "yes".

The object is set at regular intervals to the value "1" for an adjustable period and then reset to "0".

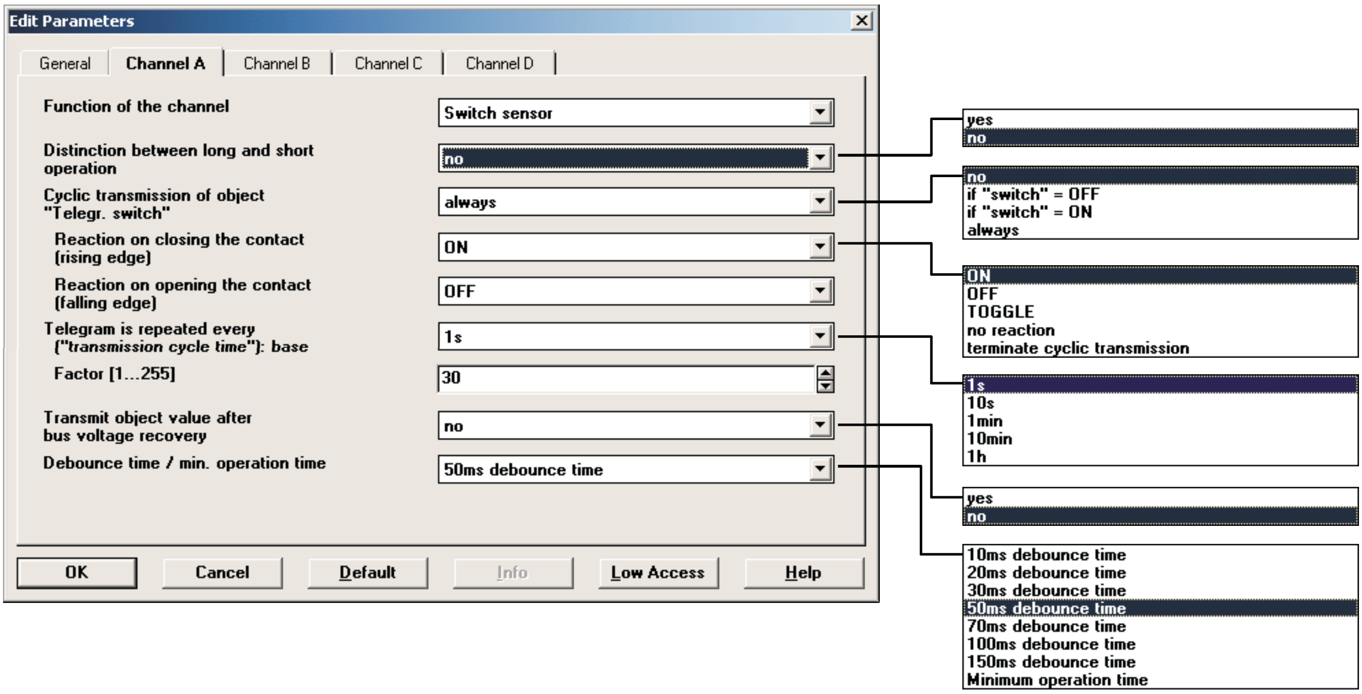
It can be used for example to trigger a valve purge at regular intervals (see "Valve purge" object).

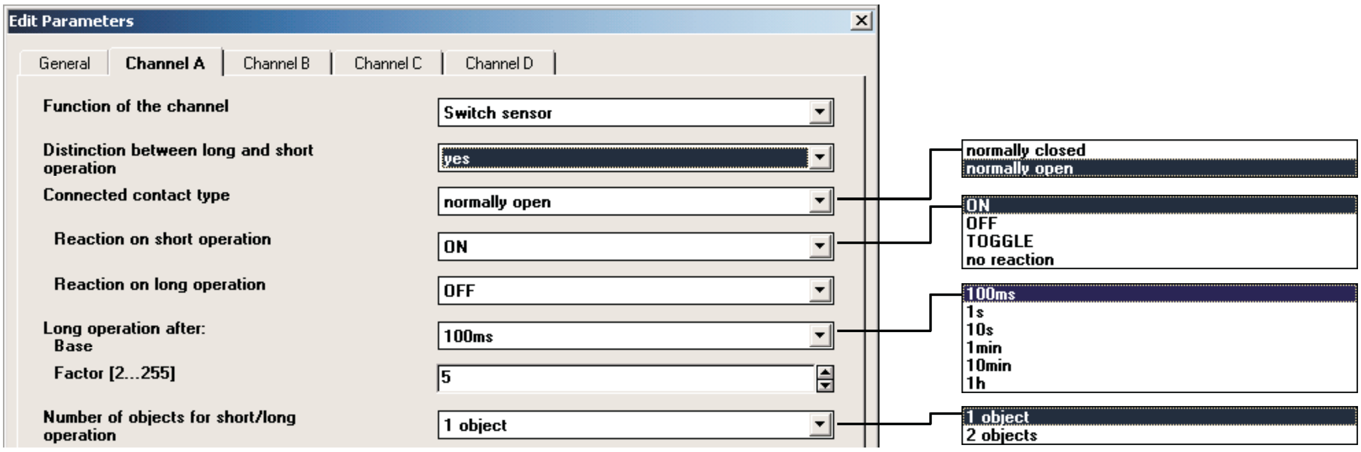
After bus voltage recovery, this object sends the value "0" to the bus and the purge cycle is restarted.

4.4 Function: "Switch sensor"

The following parameters and objects are visible if the "Switch sensor" function has been selected.

4.4.1 Parameters





Parameter: "Distinction between long and short operation"

This parameter sets whether the input distinguishes between a short and long operation. If "yes" is selected, there is a waiting period after the opening/closing of the contact to determine whether the operation is long or short. Only then is a possible reaction triggered.

The following drawing clarifies the function:

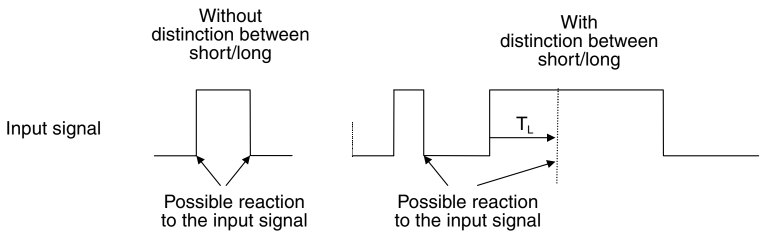


Diagram 1: Distinction between short/long operation in the “Switch sensor” function

T_L is the period after which a long operation is detected.

Parameter: “Cyclic transmission of object ‘Telegr. switch’”

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

The object “Telegr. switch” can be sent cyclically e.g. to monitor the life signs of the sensor.

If the parameter value “always” is selected, the object sends cyclically on the bus, regardless of its value.

If the parameter value “if ‘switch’ = ON” or “if ‘switch’ = OFF” is set, only the corresponding object value is sent cyclically.

You can receive further information about the topic “Cyclical sending” in section 5.3.

Parameter: “Reaction on closing the contact” or “Reaction on opening the contact”

This parameter is visible if there is no distinction between a short and long operation. It can be set separately for each pulse edge whether the object value should be “ON, “OFF”, “TOGGLE” or “no reaction”.

If cyclical sending has been parameterised, it is possible to select the option “terminate cyclic transmission” so that an operation of the input can stop the cyclical sending without a new object value being sent.

Parameter: “Telegram is repeated every (‘transmission cycle time’)”

This parameter is visible if cyclical sending has been set.

The transmission cycle time describes the interval between two telegrams that are sent cyclically.

Transmission cycle time = Base x Factor.

Parameter: “Connected contact type”

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

It is possible to select whether the contact at the input is a normally closed contact or a normally opened contact.

Parameter: "Reaction on short operation" or "Reaction on long operation"

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

It can be set for each operation at the input (short or long) how the object value is changed. The object value is updated as soon as it is established whether the operation is long or short.

Parameter: "Long operation after"

This parameter is visible if there is a distinction between a short and long operation. The period T_L is defined here, after which an operation is interpreted as "long".

T_L = Base x Factor.

Parameter: "Number of objects for short/long operation"

This parameter is visible if there is a distinction between a short and long operation.

To differentiate between short and long operations, it is possible to activate a further object by setting the parameter value "2 objects" which reacts solely to long operations.

Parameter: "Transmit object value after bus voltage recovery"

This parameter is only visible if there is no distinction between a short and long operation.

It can be set whether the current status of the input is sent on the bus (object "Telegr. switch") after bus voltage recovery (once the transmission 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.

Parameter: "Debounce time / min. operation time"

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter. A minimum operation time can only be set if there is no distinction between a short and long operation.

4.4.2 Communication objects

Object: "Telegr. switch", 1 bit (EIS 1)

According to the parameter setting, it is possible for this object to be switched on, switched off or toggled via an operation of the input.

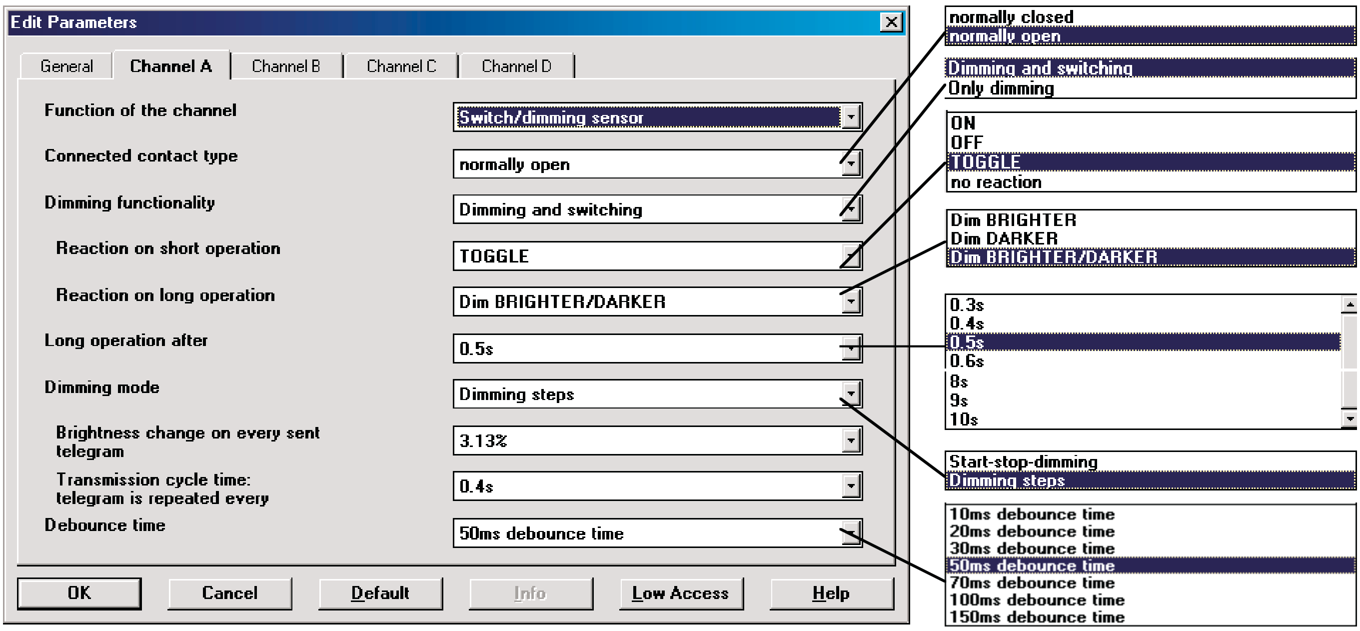
If the parameter *"Distinction between long and short operation"* is set to "yes" and the parameter *"Number of objects for short/long operation"* is set to "2 objects", two separate objects are made available. The second object is assigned to long operations.

4.5 Function: "Switch/dimming sensor"

The following section describes all the parameters and objects which are visible if the input is operated with the function "Switch/dimming sensor". The function enables the operation of dimmable lighting. 1 button operation is also possible.

Further details about the dimming function can be found in section 5.4.

4.5.1 Parameter



Parameter: "Connected contact type"

This parameter defines whether the contact at the input is a normally open contact or a normally closed contact.

Parameter: "Dimming functionality"

This parameter determines whether the lighting is only dimmed ("Only dimming") or whether it should also be switched ("Dimming and switching"). In this case, the lighting is dimmed via a long operation and switched via a short operation.

The benefit of the setting "Only dimming" is that there is no distinction between a short and long operation. The dimming command is therefore carried out immediately after the push button action; there is no delay to determine whether the operation is long or short.

Parameter: "Reaction on short operation"

This parameter is visible if the value "Dimming and switching" has been set in the parameter *"Dimming functionality"*. A short operation changes the value of the object "Telegr. switch".

This parameter sets whether the object "Telegr. switch" is toggled after a short operation (typically: dimming with 1 button) or only switched on and off (typically: dimming with 2 buttons).

Parameter: “Reaction on long operation”

This parameter is visible if the value “Dimming and switching” has been set in the parameter *“Dimming functionality”*. A long operation changes the value of the object “Telegr. dimming”.

This parameter sets whether the object “Telegr. dimming” sends a dim brighter or a dim darker telegram after a long operation. The setting “Dim BRIGHTER/DARKER” must be selected for dimming with 1 button. The opposite dimming command to the last command is sent in this case.

Parameter: “Long operation after”

This parameter is visible if the value “Dimming and switching” has been set in the parameter *“Dimming functionality”*. The period T_L is defined here, after which an operation is interpreted as “long”.

Parameter: “Reaction on operation”

This parameter is visible if the value “Only dimming” has been set in the parameter *“Dimming functionality”*. There is no distinction between a short and long operation. The meaning of the parameter settings corresponds to those of the parameter *“Reaction on long operation”* (see above).

Parameter: “Dimming mode”

Normal “Start-stop dimming” begins the dimming process with a dim darker or brighter telegram and ends the dimming process with a stop telegram. Cyclical sending of the dimming telegram is not required in this case.

For “Dimming steps”, the dimming telegram is sent cyclically during a long operation. Once the operation has finished, a stop telegram ends the dimming process.

Parameter: “Brightness change on every sent telegram”

This parameter is only visible for “Dimming steps”. It can be set, which change in brightness (percentage value) causes a dimming telegram to be sent cyclically.

Parameter: “Transmission cycle time: telegram is repeated every”

If “Dimming steps” has been set, the dimming telegram is sent cyclically during a long operation. The transmission cycle time corresponds to the interval between two telegrams during cyclical sending.

Parameter: “Debounce time / min. operation time”

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter. A minimum operation time can only be set if the value “Only dimming” has been set in the parameter *“Dimming functionality”*.

4.5.2 Communication objects

Object: “Telegr. switch”, 1 bit (EIS 1)

This object is visible if the value “Switching and dimming” has been set in the parameter *“Dimming functionality”*.

Depending on the parameter setting, the object value can be switched on, off or toggled after a short operation. For dimming with 1 button, this object should be linked with the status response of the dimming actuator as a non-sending group address. The input is thus informed about the current switching status of the dimming actuator.

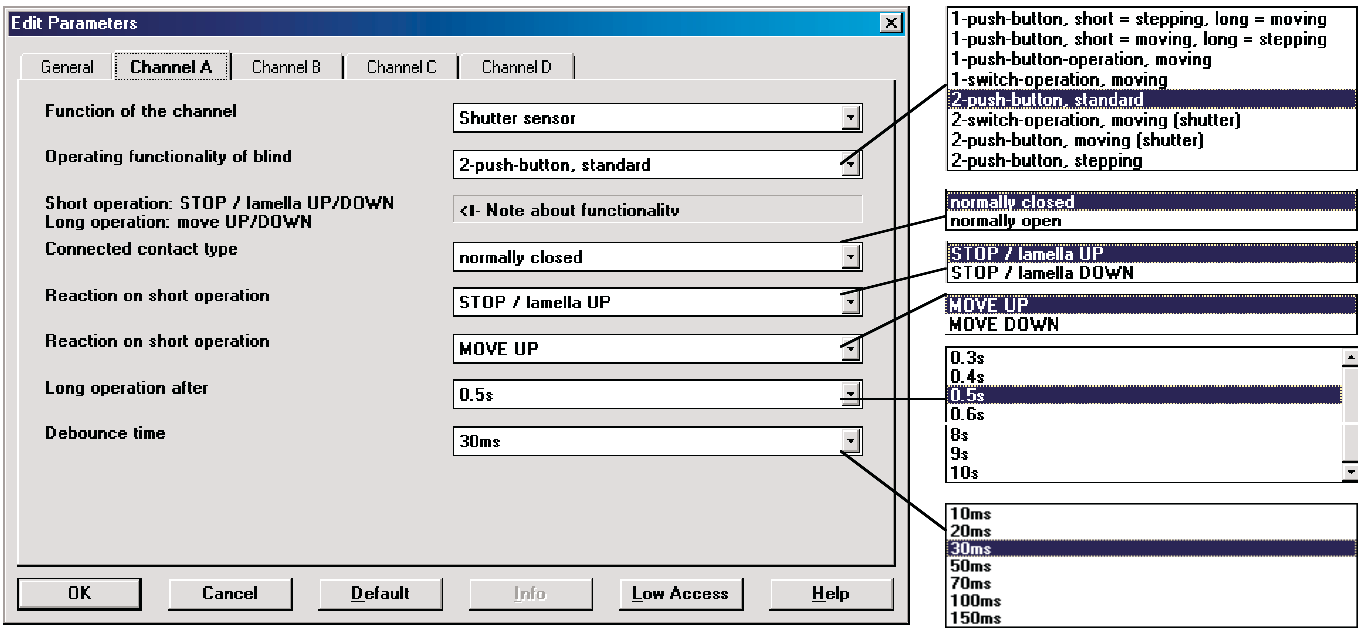
Object: “Telegr. dimming”, 4 bit (EIS 2)

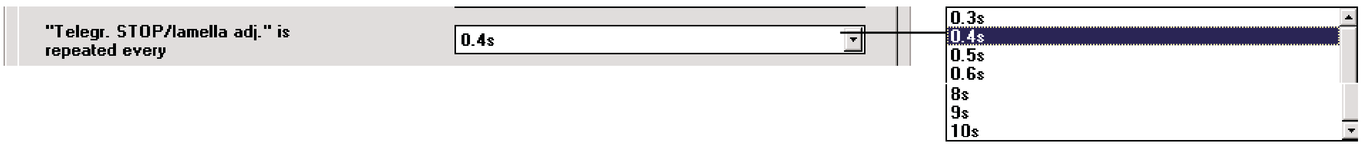
A long operation of the input causes a dim brighter or dim darker command to be sent on the bus via this object. A stop command is sent at the end of this operation.

4.6 Function: “Shutter sensor”

The following section describes all the parameters and objects that are visible if the input is operated with the “Shutter sensor” function. The function enables the operation of shutters and blinds with push buttons or switches. 1 push button operation and 1 switch operation modes are possible.

4.6.1 Parameters





Parameter: “Operating functionality of blind”

This parameter defines the type of operation. The following table 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* For reversal of lamella adjustment, 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 lamella adjustment command*

1 push button operation, moving	
On operation	The following commands are sent in sequence: … -> “MOVE UP” -> “STOP / lamella UP” -> “MOVE DOWN” -> “STOP / lamella DOWN” -> …

1 switch operation, moving	
Start of operation	Alternately “MOVE UP” or “MOVE DOWN”
End of operation	STOP / lamella adjustment*

*** Note:** If the actuator is in a limit position (see “Upper limit position” or “Lower limit position”), the direction of movement is preset.

In 1 push button/switch operation mode, the last direction of movement is determined via the last update of the object “Telegr. shutter UP/DOWN”.

2 push button, standard	
Short operation	“STOP / lamella UP” or “STOP / lamella DOWN” (can be parameterised)
Long operation	“MOVE UP” or “MOVE DOWN” (can be parameterised)

2 switch operation, moving (shutter)	
Start of operation	“MOVE UP” or “MOVE DOWN” (can be parameterised)
End of operation	“STOP / lamella UP” or “STOP / lamella DOWN”

2 push button, moving (shutter)	
On operation	The following commands are sent in sequence: … -> “MOVE UP” -> “STOP / lamella UP” -> … or “MOVE DOWN” -> “STOP / lamella DOWN” -> …

2 push button, stepping	
On operation	“STOP / lamella UP” or “STOP / lamella DOWN”

Parameter: “Connected contact type”

This parameter defines whether the contact at the input is a normally open contact or a normally closed contact.

Parameter: “Reaction on operation”

This parameter is visible if there is no distinction between a short and long operation. It can be set whether the input triggers commands for upward movement (“UP”) or downward movement (“DOWN”).

Parameter: “Reaction on short operation” or “Reaction on long operation”

This parameter is visible in all operating modes in which there is a distinction between a short and long operation. It can be set whether the input triggers commands for upward movement (“UP”) or downward movement (“DOWN”).

Parameter: “Long operation after”

This parameter is visible in all operating modes in which there is a distinction between a short and long operation. The period which defines a long operati-on is set here.

Parameter: "'Telegr. "STOP/lamella adj.' is repeated every"

This parameter is visible in operating modes in which the object "Telegr. STOP/lamella adj." is sent cyclically on the bus during a long operation. The interval between two telegrams is set here.

Parameter: "Debounce time"

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter.

4.6.2 Communications objects

Object: "Telegr. shutter UP/DOWN", 1 bit (EIS 7)

This communication object sends a shutter movement command (UP or DOWN) on the bus. The device also recognises movement commands of another sensor via the receipt of telegrams.

Telegram value	"0"	UP
	"1"	DOWN

Object: "Telegr. STOP/lamella adj.", 1 bit (EIS 7)

This communication object sends a STOP or lamella adjustment telegram.

Telegram value	"0"	STOP / lamella UP
	"1"	STOP / lamella DOWN

Object: "Upper limit position", 1 bit (EIS 1)

The shutter actuator reports via this object whether it is located in the upper limit position ("Blind open").
The object is intended for 1 push button operation.

Telegram value	"0"	No upper limit position
	"1"	Upper limit position

Object: "Lower limit position", 1 bit (EIS 1)

The shutter actuator reports via this object whether it is located in the lower limit position ("Blind closed").
The object is intended for 1 push button operation.

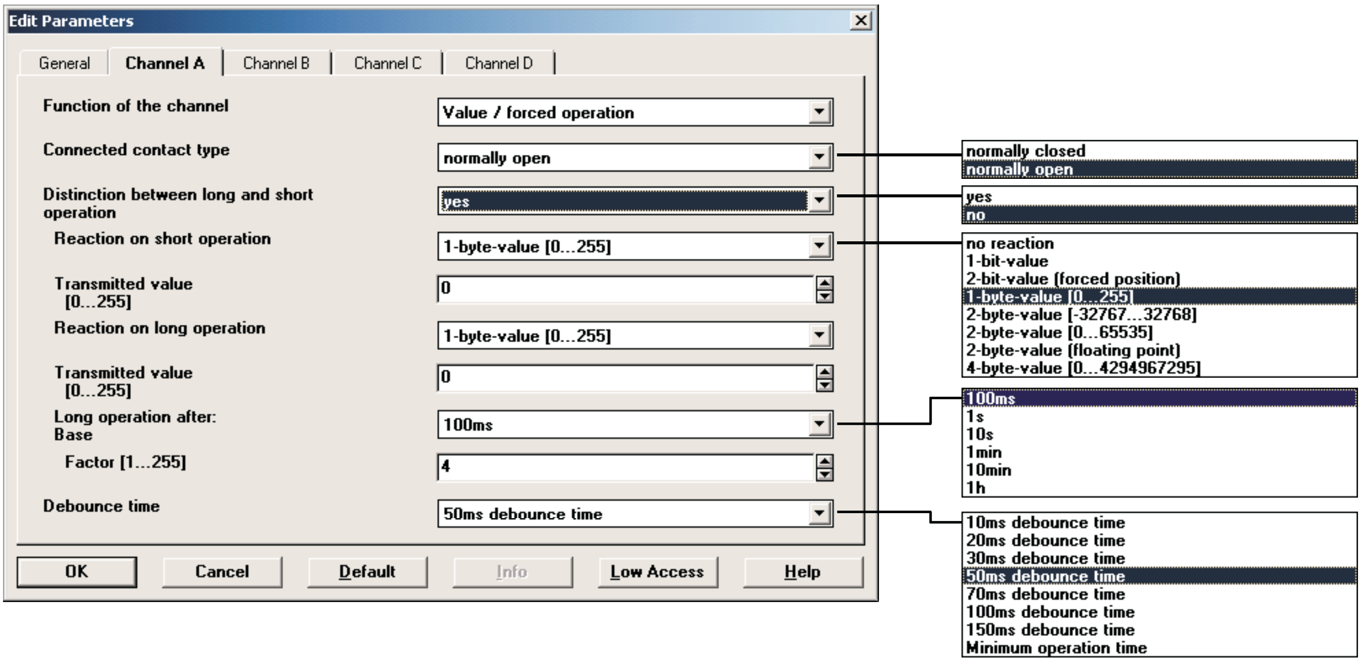
Telegram value	"0"	No lower limit position
	"1"	Lower limit position

4.7 Function: "Value / forced operation"

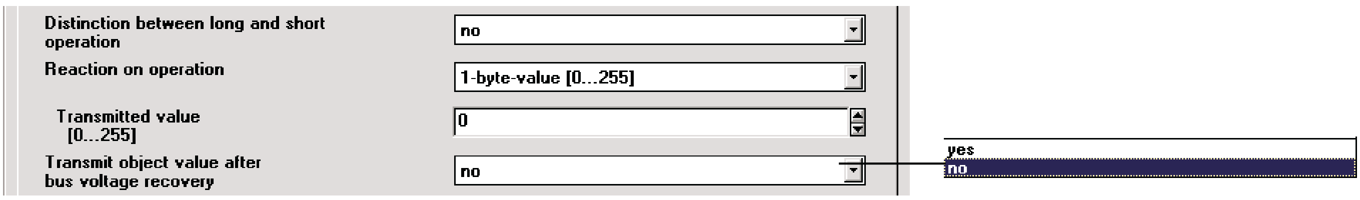
This section describes all the parameters and objects which are visible if the input is operated with the function "Value / forced operation".
The function enables the sending of values for any data types.

4.7.1 Parameters

Parameter window when there is a distinction between a short and long operation:



Parameter window when there is no distinction between a short and long operation:



Parameter: "Connected contact type"

This parameter defines whether the contact at the input is a normally open contact or a normally closed contact.

Parameter: "Distinction between long and short operation"

This parameter sets whether the input distinguishes between a short and long operation. If "yes" is selected, there is a waiting period after the opening/closing of the contact to determine whether the operation is long or short. The input then reacts accordingly.

Parameter: "Reaction on operation"

This parameter is visible if there is no distinction between a short and a long operation. It defines the data type that is sent when the contact is pressed.

Parameter: “Reaction on short operation” or “Reaction on long operation”

This parameter is visible if there is no distinction between a short and long operation. It defines the data type that is sent after a short or long operation.

Parameter: “Transmitted value”

This parameter defines the value which is sent on operation. The value range is dependent on the selected data type. Two values can be set here when there is a distinction between a short and long operation.

Parameter: “Long operation after”

This parameter is visible if there is a distinction between a short and long operation. The period T_L is defined here, after which an operation is interpreted as “long”.

T_L = Base x Factor

Parameter: “Transmit object value after bus voltage recovery”

This parameter is visible if there is no distinction between a short and long operation. If “yes” is selected, the device sends the object “Telegr. value” on the bus after bus voltage recovery (once the transmission delay has elapsed).

Parameter: “Debounce time / min. operation time”

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter. A minimum operation time can only be set if there is no distinction between a short and long operation.

4.7.2 Communication objects

The following tables provides an overview of the available data types:

Data width, type	Value range	EIS type	Typical application
1 bit	0, 1	EIS 1	Switching command
2 bits	0, 2, 3	EIS 8	Forced operation
1 byte without sign	0 … 255	EIS 6	Brightness value, position value
2 bytes, integer with sign	– 32768 … +32767	EIS 10	Count value
2 bytes, integer without sign	0 … 65535	EIS 10	Count value
2 bytes, floating point value*	– 100 … +100	EIS 5	Temperature values
4 bytes, integer without sign	0 … 4294967295	EIS 11	Count value

* sends values with the firm exponent of 3

Object: “Telegr. value (…)” (various types)

This communication object sends a value on the bus when the contact is opened or closed. The value and data types can be freely selected in the parameters.

When there is a distinction between a short and long operation, 2 objects are visible per input. One object only transmits a value after a short operation while the other object only sends after a long operation.

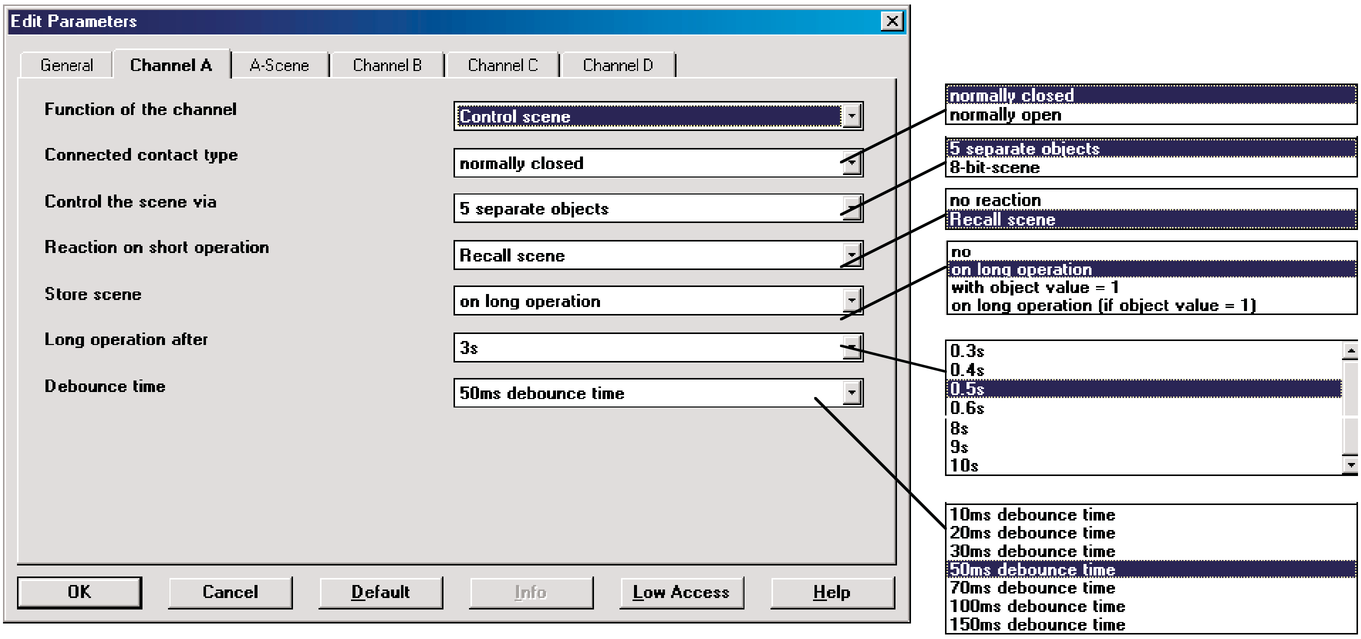
Note: By default, the “Write” flag is deleted for the value objects (except for the 1 bit objects). The object value cannot thus be modified via the EIB. If this function is required, the “Write” flag must be set in ETS. On bus voltage recovery, the object value is overwritten with the parameterised value.

4.8 Function: “Control scene”

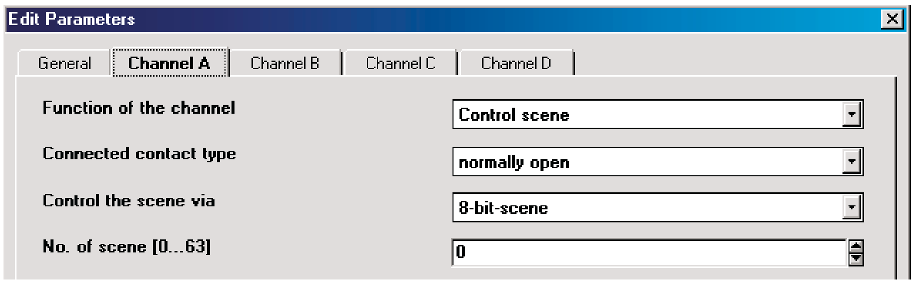
The following section describes all the parameters and objects that are visible when the input is operated with the function “Control scene”. This function enables the states of several actuator groups to be recalled and stored.
A detailed explanation of the function can be found under section 5.5.

4.8.1 Parameters

Parameter window for scene control via “5 separate objects”:

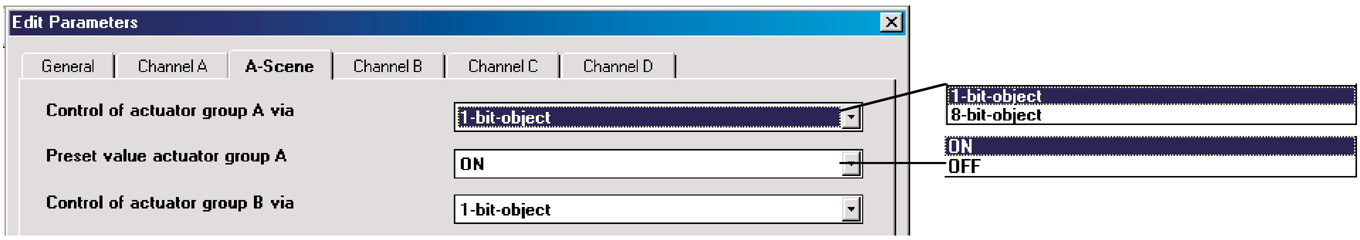


Parameter window for scene control via “8 bit scene”:



(Remaining parameters as above)

Additional parameter window “A-Scene” for controlling the scene via “5 separate objects”:



Parameter: “Connected contact type”

This parameter defines whether the contact at the input is a normally open contact or a normally closed contact.

Parameter: "Control the scene via"

It is possible to select whether the scene control is carried out via "5 separate objects" or whether values that are stored in the actuators are recalled and saved via an "8 bit scene" (see section 5.5 for further information).

Parameter: "Reaction on short operation"

This parameter defines whether a short operation of the input causes a lightscene to be recalled or no reaction takes place.

Parameter: "Store scene"

This parameter specifies how the saving of the current scene can be triggered and defines the function of the object "Store scene". This is dependent on the type of scene control. The following table provides an overview:

Control the scene via **"5 separate objects"**

Parameter value	Behaviour
"on long operation"	As soon as a long operation is detected, the object "Store scene" sends the value "1" on the bus and the object values "Telegr. switch/value actuator group A…E" are read out via the bus and stored in the object values. The objects "Telegr. switch/value actuator group A…E" can be modified via the bus for the duration of the long operation. Once the long operation has finished, the object "Store scene" sends the value "0" on the bus and the current object values are stored in the device.
"if object value = 1"	If the object "Store scene" receives the value "1", the object values "Telegr. switch/value actuator group A…E" are read out via the bus. While the object value is "1", the objects "Telegr. switch/value actuator group A…E" can be modified via the bus. On receipt of the object value "0", the current object values are stored in the device. **Important:** The storing of the current scene requires the object values "1" and "0" to be sent in succession.
"on long operation (if object value = 1)"	If the object "Store scene" receives the value "1" on the bus, the next long push button action leads to the value "1" being sent via the object "Store scene". The scanning of the object values "Telegr. switch/value actuator group A…E" is then carried out. After the end of the long operation, the object values "Telegr. switch/value actuator group A…E" are stored in the device. Provided that a "1" has not been received at the object "Store scene", a long operation is interpreted in the same way as a short operation. The same applies if the object "Store scene" has received the value "0".

Control the scene via **“8 bit scene”**

Parameter value	Behaviour
“on long operation”	After a long operation, the object “8 bit scene” sends a save command on the bus and thereby triggers the storing of the current scene in the actuators. The object “Store scene” has no function.
“if object value = 1”	If the object “Store scene” receives the value “1”, the object “8 bit scene” sends a save command on the bus.
“on long operation (if object value = 1)”	If the object “Store scene” receives the value “1” on the bus, the next long push button action triggers the sending of a save command via the object “8 bit scene”. Provided that a “1” has not been received at the object “Store scene” since the last save, a long operation is interpreted in the same way as a short operation. The same applies if the value “0” has been received.

Parameter: “Long operation after”

This parameter is visible if the saving of the scene is possible via a long operation. The period is defined here, after which an operation is interpreted as “long”.

Parameter: “Debounce time”

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter.

Parameter window: ***“A-Scene”***

This parameter window is visible if the control of the lightscenes is carried out via “5 separate objects”.

Parameter: “Control of actuator group A…E via”

It can be set for each actuator group whether the control is carried out via a “1 bit object” or an “8 bit object”. The type of the communication object “Telegr. switch/value actuator group A…E” is set accordingly.

Parameter: “Preset value actuator group A…E”

A value is preset for each actuator group A…E in this parameter. If a scene has been stored, the current object values of actuator groups A…E are overwritten with the values set here following programming or bus voltage recovery and when the scene is recalled again.

4.8.2 Communication objects

Object: “Telegr. switch actuator group A … E”, 1 bit (EIS 1) or “Telegr. value actuator group A … E”, 8 bit (EIS 6)

These objects are visible if the scene is controlled via “5 separate objects”.

They control several 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.

Object: "8 bit scene", 8 bit

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

M:	0 – Scene is recalled
	1 – Scene is stored
x:	Not used
S:	Number of the scene (0…63)

Object: "Store scene", 1 bit (EIS 1)

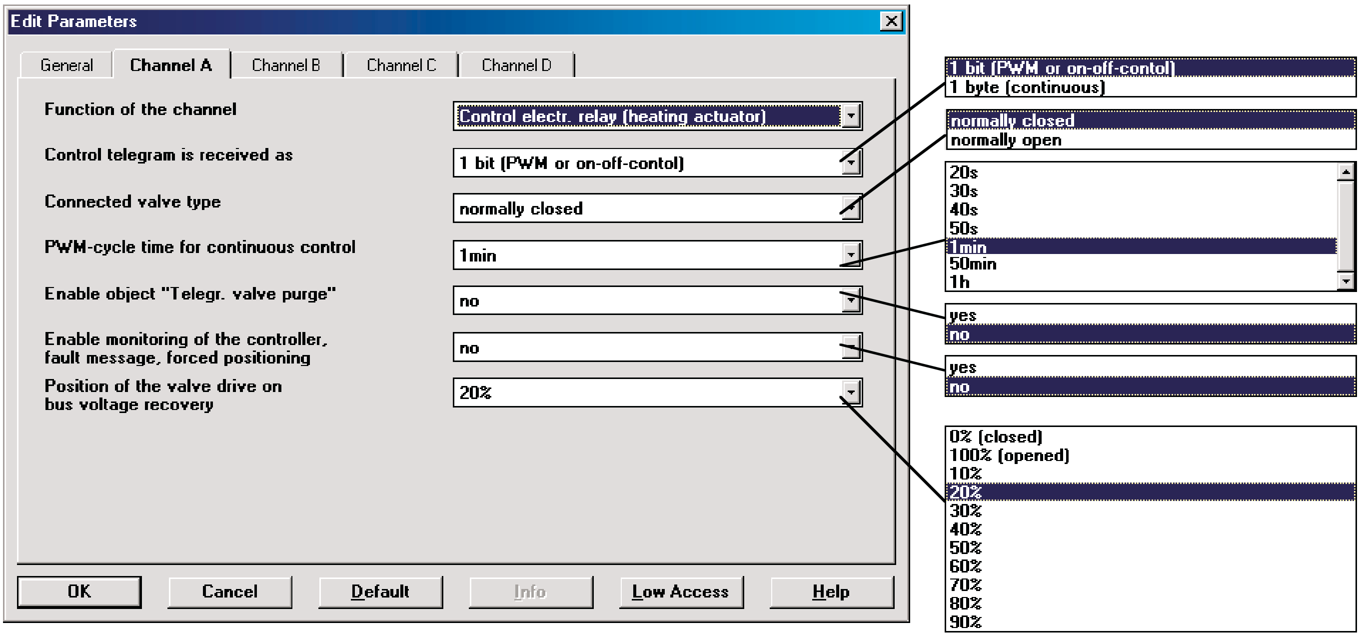
This object can be used to trigger the saving of a scene via the bus or to indicate that the scene has been stored. The function depends on the method of storing the scene.

Refer to the description of the parameter *"Store scene"* for more detailed information.

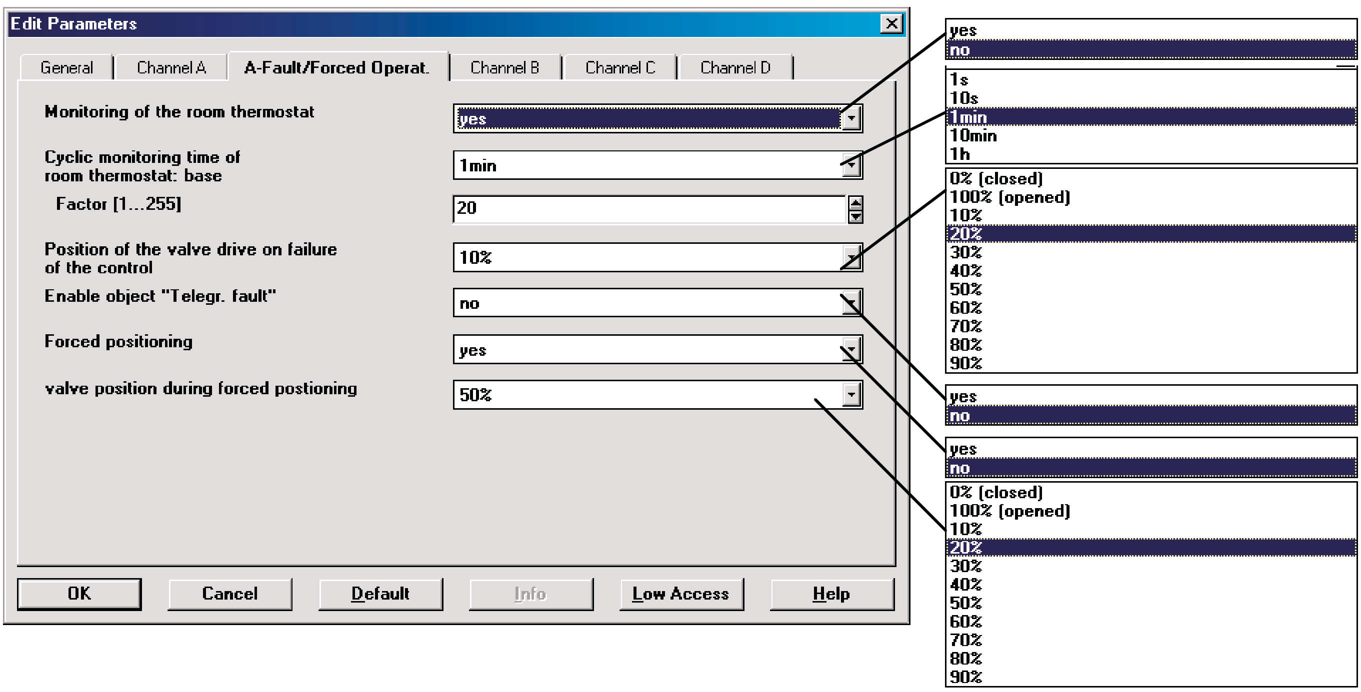
4.9 Function: “Control electronic relay (heating actuator)”

The following section describes all the parameters and objects that are visible if the input is operated with the function “Control electronic relay”. A thermal valve drive for heating valves can be controlled via an electronic relay. Refer to section 5.6 for a more detailed explanation of the function.

4.9.1 Parameters



Additional parameter window when the parameter *Enable monitoring of the controller, fault message, forced positioning* is set to “yes”:



Parameter: “Control telegram is received as”

The heating actuator can either be controlled via the 1 bit object “Telegr. 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 regulates the heating actuator via normal switching commands. It is thus possible to implement a simple 2-step closed-loop control or a pulse width modulation of the control value.

In the case of **1 byte** control, a value of 0…255 (corresponds to 0%…100%) is preset by the room thermostat. This function is usually referred to as “continuous-action control”. The valve is closed at 0% and fully opened at 100%. The heating actuator controls intermediate values via pulse width modulation (see the graphic is section 5.6).

Parameter: “Connected valve type”

In this parameter, it is possible to set whether a valve is “de-energised closed” or “de-energised opened”. In the case of “de-energised closed”, the opening of the valve is achieved by closing the electronic relay while the process is reversed for “de-energised opened”.

Parameter: “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.

For 1 bit control and 1 byte control, this period is only used when the actuator is controlled in fault mode, during forced positioning and directly after bus voltage recovery.

Parameter: “Enable object ‘Telegr. valve purge’”

The object “Valve purge” is enabled with this parameter.

Parameter:
“Enable monitoring of the controller, fault message, forced positioning”

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.

Parameter: “Position of the valve drive on bus voltage recovery”

This parameter defines how the valve drive is controlled after bus voltage recovery, until the first switching or positioning command of the room thermostat is received. The parameterised value is selected as the PWM cycle time.

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

This parameter window is visible if the value “yes” is entered in the parameter *“Enable monitoring of the controller, fault message, forced positioning”*.

Parameter: “Monitoring of the room thermostat”

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

The telegrams of the room thermostat are transmitted to the electronic actuator at specific cyclic intervals. If one or more of these telegram sequences is omitted, there may be a communications fault or a defect in the room thermostat. If no telegrams are sent to the objects “Telegr. switch” or “Control vale (PWM)” for the duration of the **cyclic monitoring time**, the actuator switches to fault mode and triggers a safety position. The fault mode is finished as soon as a telegram is received again.

Parameter: “Cyclic monitoring time of room thermostat”

The cyclic monitoring time for the telegrams of the room thermostat is set in this parameter.

Period = Base x Factor

Parameter: “Position of the valve drive on failure of the control”

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

Parameter: “Enable object ‘Telegr. fault’”

The object “Telegr. fault” can be enabled in this parameter. It has the object value “ON” during fault mode. 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.

Parameter: “Forced positioning”

This parameter enables the forced positioning function. During forced positioning, the actuator triggers a freely selectable forced positioning. This has the highest priority i.e. it is not modified by a valve purge or a safety position. The forced positioning can be activated via the object “Forced positioning” = “ON” and deactivated via “Forced positioning” = “OFF”.

Parameter: “Valve position during forced positioning”

In this parameter, the valve position triggered by the actuator is defined during the forced positioning. The PWM cycle time T_{CYC} of the control is defined in the parameter *“Cycle time for continuous control”*.

4.9.2 Communication objects

Object: “Telegr. switch”, 1 bit (EIS 1)

This object is visible if the control of the heating actuator 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”.

Telegram value:	“0”	Close valve
	“1”	Open valve

Object: “Control value (PWM)”, 8 bit (EIS 6)

This object is visible if the control of the heating actuator is carried out via an 8 bit object e.g. during continuous control. The object value (0…255) determines the selection ratio (mark-to-space ratio) of the valve.

Telegram value:	“0”	Close valve
	“...”	Mark-to-space ratio
	“255”	Open valve

Object: “Valve purge”, 1 bit (EIS 1)

This object is visible if the parameter *“Enable object ‘Telegr. 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”.

Telegram value:	“0”	Stop valve purge
	“1”	Start valve purge

Object: “Forced positioning”, 1 bit (EIS 1)

This object is visible if 1 bit forced positioning is enabled in the parameters.

The forced positioning of the device is activated or deactivated via this object. In this way, the valve can be controlled with a defined value. Forced positioning has the highest priority.

Telegram value:	“0”	Stop forced positioning
	“1”	Start forced positioning

Object: “Telegr. status/ackn.”, 1 bit (EIS 1)

This object reports the switching state of the heating actuator.
The object value is sent after each change of the output.

Telegram value:	“0”	Valve is closed
	“1”	Valve is opened

Note: For PWM continuous control, this object is sent after each change in the output. The additional telegram load should therefore be taken into account particularly if a short PWM cycle time has been set.

Object: “Telegr. fault”, 1 bit (EIS 1)

This object is visible if the fault message has been enabled in the parameters.

If the output does not receive any telegrams from the room thermostat via the object “Telegr. switch” or “Control value (PWM)” for an adjustable period, the devices switches to fault mode and reports this via the object.

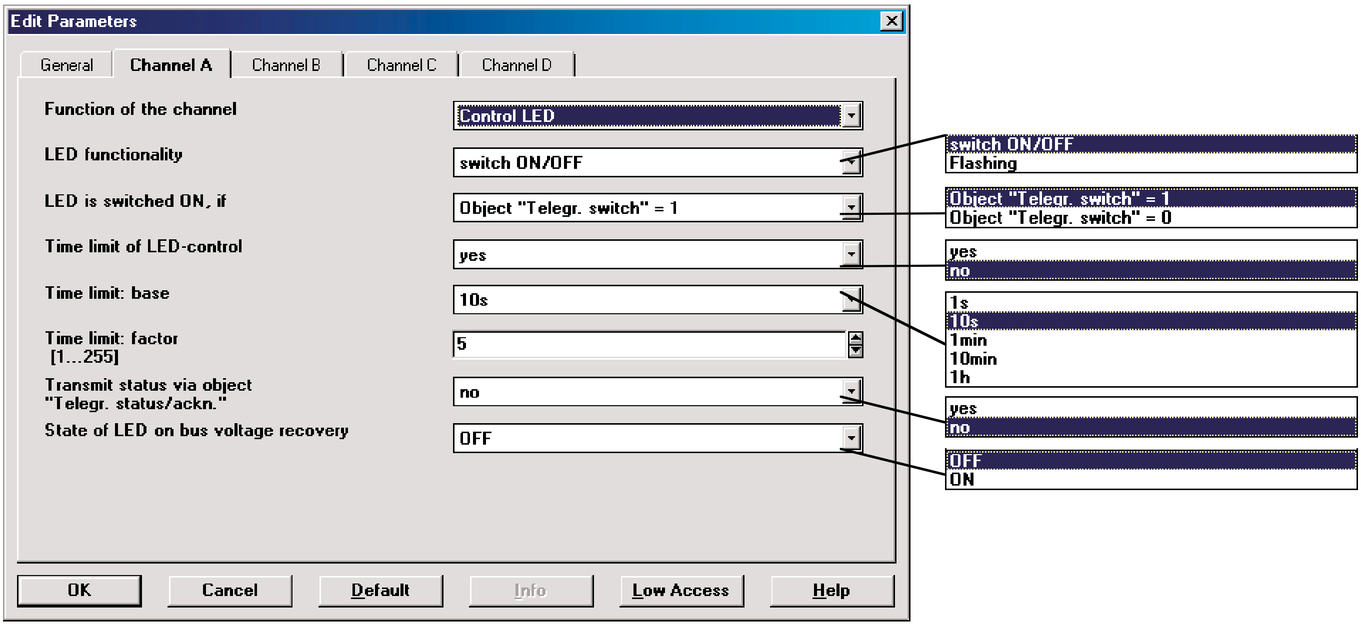
Telegram value:	“0”	No fault
	“1”	Fault mode active

4.10 Function: “Control LED”

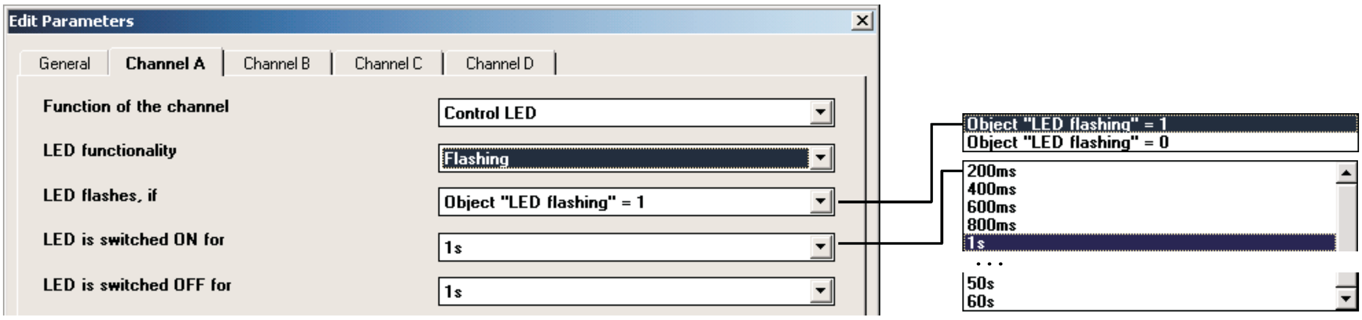
The following section describes all the parameters and objects which are visible if the input is operated with the function “Control LED”. It is possible for example to confirm an operation via an LED.

4.10.1 Parameters

Parameter window for *LED function* = “switch ON/OFF”



Parameter window for *LED function* = “Flashing”



Parameter: “LED functionality”

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

Parameter: “LED is switched ON, if”

This parameter is visible if the LED function has been set to “switch ON/OFF”. It can be defined which state the object “LED, switching” must have so that the LED is switched on.

Parameter: “LED flashes, if”

This parameter is visible if the LED function “Flashing” has been set.
It can be defined which state the object “LED, flashing” must have so that the flashing of the LED is active.

Parameter: "LED is switched ON for" or "LED is switched OFF for"

This parameter is visible if the LED function "Flashing" has been set.

It is defined how long the LED is switched on or switched off during the flashing signal. The flash rate of the signal can thus be set.

Parameter: "Time limit of LED control"

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

Parameter: "Time limit" (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 or flashes. Once this time limit has elapsed, the LED is switched off.

Period = Base x Factor

Parameter: "Transmit status via object 'Telegr, status/ackn.'"

The object "Telegr. status/ackn." Is enabled via this parameter. It indicates with the value "ON" that the LED has been switched on or is flashing.

4.10.2 Communication objects

Object: "LED, switching", 1 bit (EIS 1)

This object is visible if the parameter *"LED function"* has been set to "switch ON/OFF". The object switches the LED on and off.
The telegram values can be set in the parameters.

Object: "LED, flashing", 1 bit (EIS 1)

This object is visible if the parameter *"LED function"* has been set to "Flashing". The flashing of the LED can be started and stopped via this object.

Telegram value:	"0"	Stop flashing
	"1"	Start flashing

Object: "LED, permanent ON", 1 bit (EIS 1)

This object is visible if the parameter *"LED function"* has been set to "Flashing".

The LED can be switched on permanently via this object.
The flashing function is deactivated in this way.

Telegram value:	"0"	Flashing function active
	"1"	LED permanently ON

Object: "Telegr. status/ackn.", 1 bit (EIS 1)

This object is visible if the value "yes" has been set in the parameter *"Transmit status via …"*. It reports the status of the output.

Telegram value:	"0"	LED is switched off
	"1"	LED is switched on or flashes

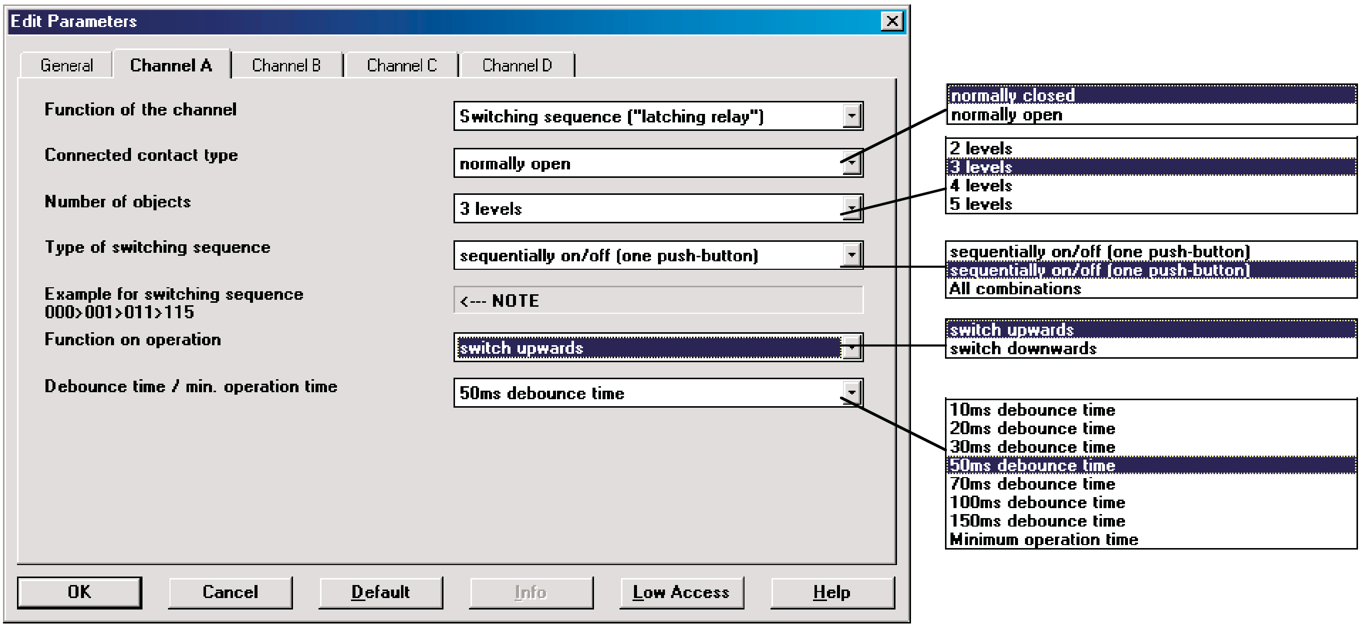
4.11 Function: “Switching sequence”

The following section describes all the parameters which are visible if the input is operated with the function “Switching sequence”. A switching sequence enables the stepwise modification of several values via a single operation.

Next switching level = actual value of objects ± 1

+ 1 → Switch upwards
– 1 → Switch downwards

4.11.1 Parameters



Parameter: “Connected contact type”

This parameter defines whether the contact at the input is a normally open contact or a normally closed contact.

Parameter: “Number of objects”

The number of levels (max. 5) is identical to the number of communication objects: objects “Value 1” to “Value n” are enabled.

Parameter: “Type of switching sequence”

The switching sequence can be selected here. Each sequence has other object values for each switching level. The following switching sequences are possible (a detailed description can be found in section 5.7):

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

The example is based on the status of three objects (“0” = OFF, “1” = ON).

A table of the grey code can be found in section 7.1.

Parameter: “Function on operation”

Only visible in the switching sequence “sequentially on/off (several push buttons)”. It can be set whether an operation of the push button switches up or down a level.

Parameter: “Debounce time / min. operation time”

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter.

4.11.2 Communication objects

Objects: “Telegr. value 1” to “Telegr. value 5”, 1 bit (EIS 1)

The number of these objects (max. 5) is set in the parameter *“Number of objects”*. The objects represent the values within a switching sequence.

Object: “Level increment/decrement”, 1 bit (EIS 1)

On receipt of an ON telegram at this communication object, the input switches up one level in the switching sequence. On receipt of an OFF telegram, it switches down one level.

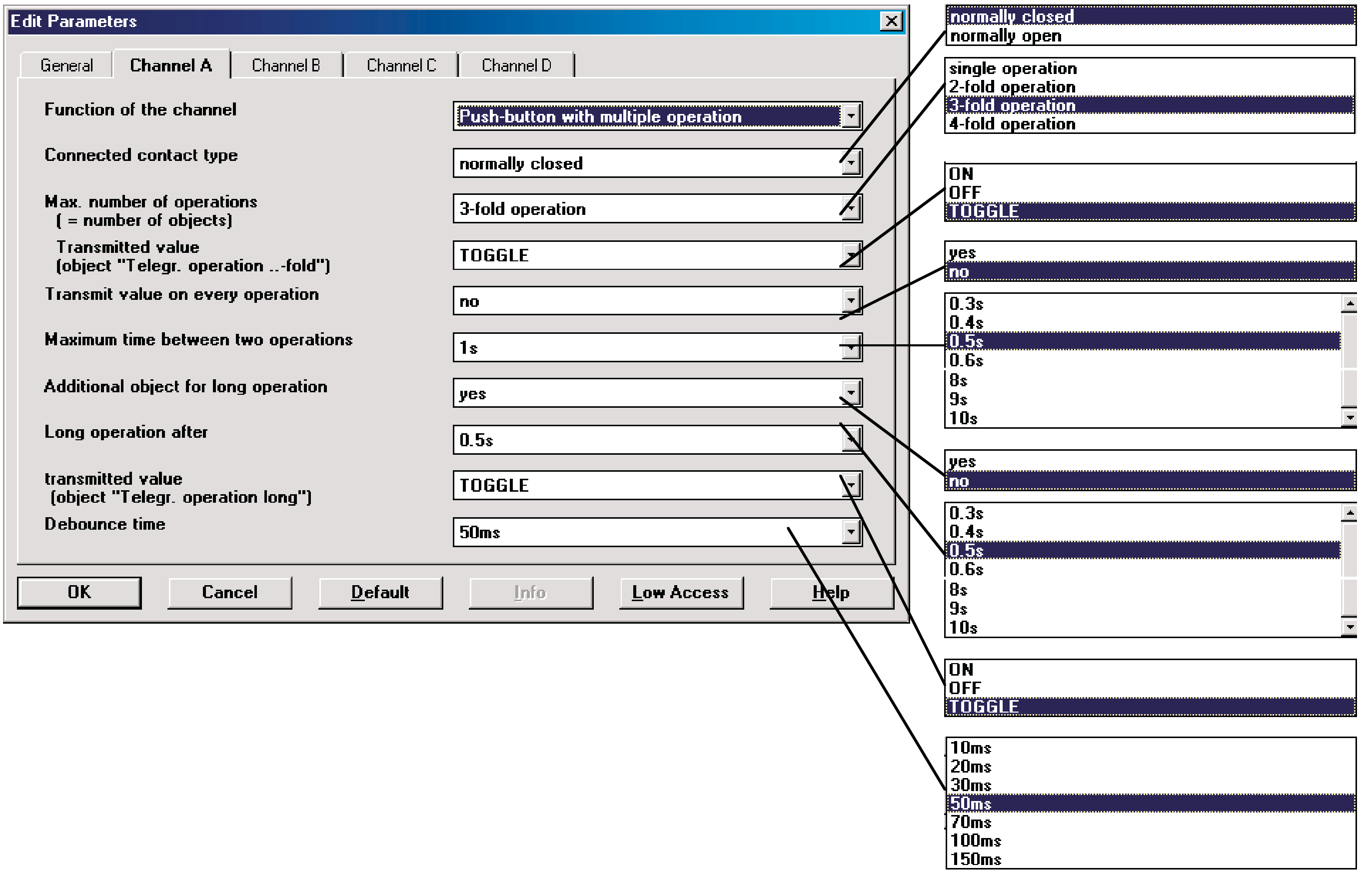
Telegram value:	“0”	Switch up one level
	“1”	Switch down one level

4.12 Function: “Push button with multiple operation”

The following section describes all the parameters and objects which are visible if the input is operated with the function “Push button with multiple operation”.

If the input is operated several times within a certain period, a specified object value can be modified depending on the number of operations. This enables e.g. different lightscenes to be implemented with multiple push button actions.

4.12.1 Parameters



Parameter: “Connected contact type”

This parameter defines whether the contact at the input is a normally open contact or a normally closed contact.

Parameter: “Max. number of operations”

This parameter specifies the maximum permitted number of operations. This number is identical to the number of communication objects “Telegr. operation x-fold”. If the actual number of operations is higher than the maximum value set here, the input reacts as if the number of operations were identical to the maximum value set here.

Parameter: “Transmitted value”

It can be set here which object value should be sent. The settings “ON”, “OFF” and “TOGGLE” are possible. The current object value is inverted in the “TOGGLE” setting.

Parameter: "Transmit value on each operation"

If "yes" is entered in this parameter, the associated object value is updated and sent after each operation in the case of multiple push button actions.

Example: For three-fold operations, the objects "Telegr. operation 1-fold" (after the first operation), "Telegr. operation 2-fold" (after the second operation) and "Telegr. operation 3-fold" (after the third operation) are sent.

Parameter: "Maximum time between two operations"

This parameter sets the interval between two operations. After an operation, there is a delay for the duration specified here. If there are no further operations within this period, the object "Telegr. switch" is sent and the period restarts after the next operation.

Parameter: "Additional object for long operation"

After a long operation of the input, a further function can be executed via the object "Telegr. switch-long". If a long operation is carried out after one or several short operations within the maximum period, the short operations are ignored.

Parameter: "Long operation after"

The period is defined here, after which an operation is interpreted as "long".

Parameter: "Transmitted value"

It can be set here whether the object value "Telegr. switch-long" should be switched on, switched off or toggled after a long operation.

Parameter: "Debounce time / min. operation time"

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter. A minimum operation time can only be set when there is no distinction between a short and a long operation.

4.12.2 Communication objects

Objects: "Telegr. operation 1-fold" to "Telegr. operation 4-fold", 1 bit (EIS 1)

The number of these objects (max. 4) is set in the parameter *"Max. number of operations".*

After multiple operation of an input, the corresponding object is sent according to the number of operations. The telegram value can be set in the parameters.

Object: "Telegr. operation-long", 1 bit (EIS 1)

This object is visible if the value "yes" has been set in the parameter *"Additional object for long operation"*.

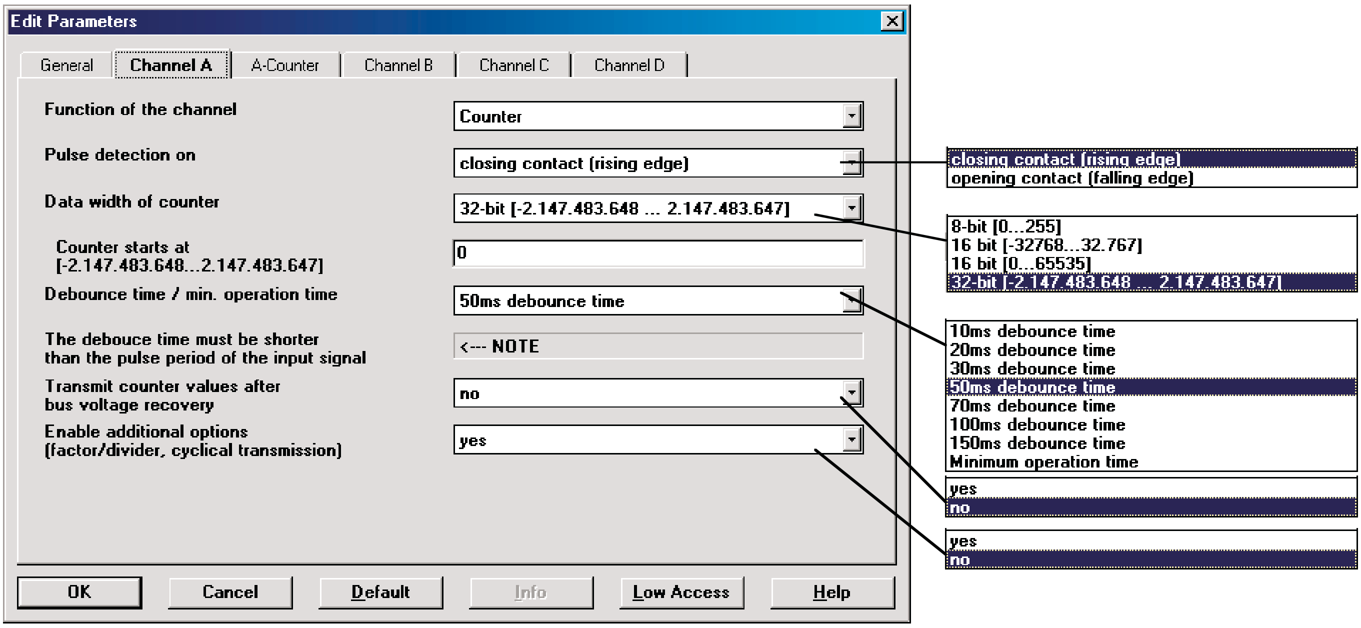
This object is sent once a long operation has been detected. The telegram value can be set in the parameters.

4.13 Function: “Counter”

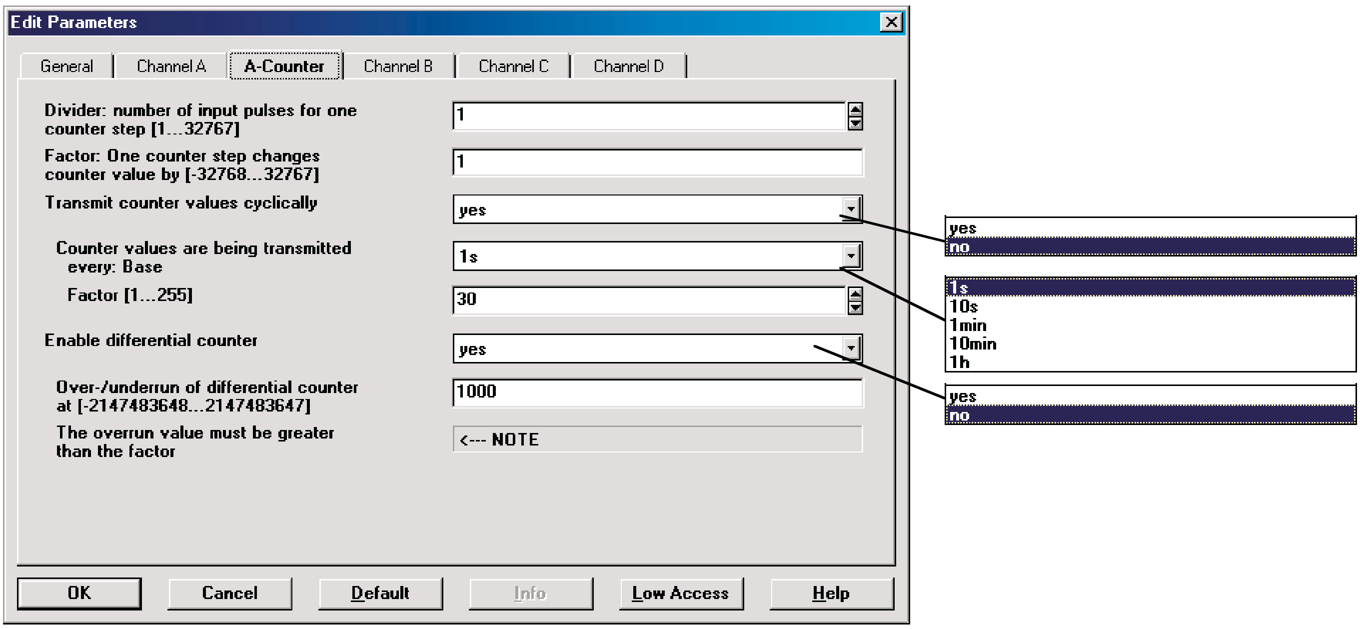
The following section describes all the parameters and objects which are visible when the input is operated with the function “Counter”.

Using the “Counter” function, the device is able to count the number of pulse edges at the input. A “differential counter” is therefore available if required in addition to the standard counter. Both counters are triggered by counting pulses but otherwise operate independently of each other. The counter always has the same data width as the differential counter.

4.13.1 Parameter



Additional parameter window if *“Enable additional options (…)”* is set to “yes”.



Parameter: "Pulse detection on"

The type of input signal is defined in this parameter. It can be set whether the contact is a normally open contact or a normally closed contact.

Parameter: "Data width of counter"

The data type of the counter (absolute counter and differential counter) is defined in this parameter. The data type specifies the counting range for the counter.

The type of the objects "Telegr. counter value …" and "Differential counter …" is adapted to the data width.

Parameter: "Counter starts at …"

The starting value of the absolute counter is defined in this parameter. The starting value is used when there is a counter overflow in order to calculate the new counter value.

Parameter: "Debounce time / min. operation time"

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 5.1 for the exact function of this parameter.

Parameter: "Transmit counter values after bus voltage recovery"

If this parameter has the value "yes", the current value of the counter is sent on the bus after bus voltage recovery (once the transmission delay has elapsed). If the differential counter has been enabled, it is also sent on the bus.

After an extended bus voltage failure, the counter is reset to the starting value. If the differential counter has been enabled, it is reset to zero. If no data loss has occurred after a short bus voltage failure, the counter contents are retained.

Parameter: "Enable additional options (…)"

If this parameter is set to "yes", the parameter window *"A-Counter"* is displayed. Additional functions are possible here.

Parameter window: "A-Counter"

Additional functions can be activated in this parameter window for the pulse counter.

Parameter: "Divider: number of input pulses for one counter step"

It can be set via this parameter how many pulses are necessary to generate a counting pulse. It thus functions as a divider.

Parameter: "Factor: one counter step changes counter value by"

This parameter defines how much the counter and differential counter should be increased by in the event of a counting pulse. It thus functions as a factor.

Parameter: "Transmit counter values cyclically"

If this parameter has the value "yes", the values of the counter and the differential counter are sent cyclically on the bus.

Parameter: "Counter values are being transmitted every"

This parameter is visible if the parameter *"Transmit counter values cyclically"* has been set to "yes". It can be set in which intervals the values are sent cyclically on the bus.

Parameter: "Enable differential counter"

The object "Differential counter" is made visible via this parameter.
The differential counter can e.g. take over the function of a daily counter.

Parameter: "Over-/underrun of differential counter at"

This parameter is visible if the parameter *"Enable differential counter"* is set to "yes".

It can be set in this parameter which value generates an overflow of the differential counter. In the event of an overflow, the same rules apply as for the standard counter. The object "Differential counter overflow" is sent in this case.

4.13.2 Communication objects

Object: "Telegr. counter value … bytes", 1 to 4 bytes

This object contains the absolute counter content of the pulse counter.
The counter can have a data width of 1 byte, 2 bytes and 4 bytes.

The following table provides an overview of the data types:

Data width	EIS type	Value range
1 byte	EIS 14	0...255
2 bytes	EIS 10	-32.768...32.767
2 bytes	EIS 10	0...65.535
4 bytes	EIS 11	-2.147.483.648...2.147.483.647

Object: "Differential counter … bytes", 1 to 4 bytes

This object is visible if the value "yes" has been set in the parameter *"Enable differential counter"*.

The object contains the status of the differential counter which is identical to the absolute counter in its counting function. In contrast to this counter however, it can be reset (object "Reset differential counter") and a counter overflow can be reported on the bus (object "Differential counter overflow"). Daily consumption values for example can be measured via the differential counter.

As soon as the differential counter reaches, exceeds or falls below the overflow value defined in the parameter *"Over-/underrun of differential counter at"*, the overflow value is deducted from the value of the differential counter.

Object: "Request counter values", 1 bit (EIS 1)

The values of the absolute counter and the differential counter are requested via this object.

Telegram value:	"0"	No reaction
	"1"	Send counter values

Object: "Differential counter overflow", 1 bit (EIS 1)

This object is visible if the value "yes" has been selected in the parameter *"Enable differential counter"*.

As soon as the value of the differential counter has exceeded or fallen below the overflow value defined in the parameter *"Over-/underrun of differential counter at"*, the object is sent on the bus (telegram value = "1").

Object: "Reset differential counter", 1 bit (EIS 1)

This object is visible if the value "yes" has been set in the parameter *"Enable differential counter"*. The differential counter can be reset to the value "0" via this object.

Telegram value:	"0"	No reaction
	"1"	Reset differential counter

4.14 Programming

The device can be programmed via the EIB Tool Software ETS2 **V1.2a** or higher. To reduce the programming time of the device via ETS, it is supplied as pre-programmed. During the programming, it is automatically detected whether the device already has the correct application program.

If the device has been pre-programmed with another version, which should only happen in exceptional cases, a full download is required. This can take several minutes.

Note: If a programmed application program needs to be reprogrammed, the device must first be unloaded via the ETS. This can be necessary in rare cases e.g. if an error occurs during a download.

Important: If a device has no function after programming, please make a new product import of the Universal Interface (.VD2-file) into the ETS and repeat the programming.

5 Special functions

The following section outlines special functions which could not be described in connection with the parameters and objects due to lack of space.

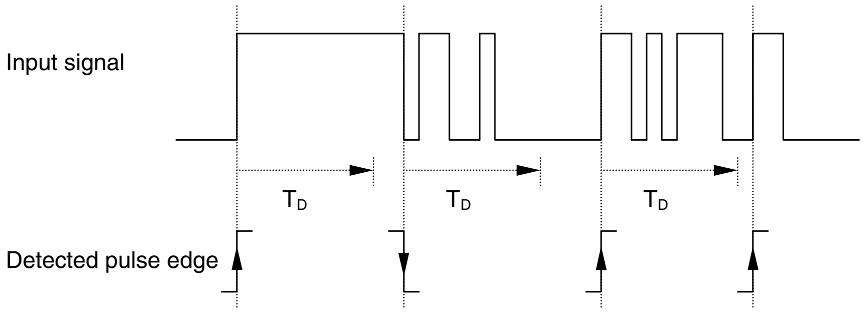
5.1 Debounce time and minimum operation time

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

Debounce time

If a pulse edge is detected at the input, the input reacts to it immediately (e.g. with the sending of a telegram). The debounce time T_D starts at the same time. During the debounce time, the signal is not evaluated at the input.

The following example clarifies this:



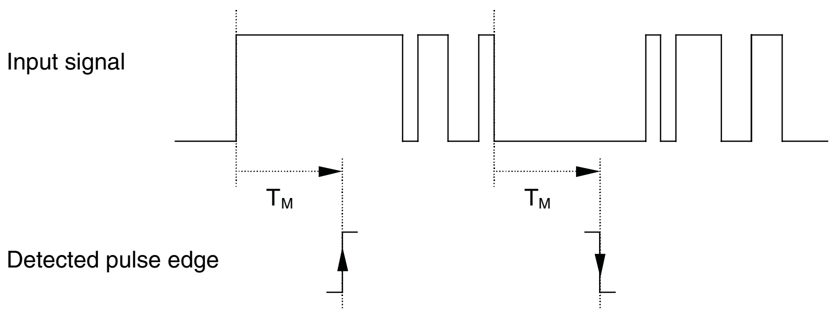
When a pulse edge is detected at the input, further pulse edges are ignored for the duration of the debounce time T_D.

Minimum operation 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 a pulse edge is detected at the input, the minimum operation time starts. No telegrams are sent on the bus at this point. The signal at the input is observed during the minimum operation time. If a further pulse edge occurs at the input during this period, it is interpreted as a new operation and the minimum operation time is restarted. If the input signal does not change during the minimum operation time, a pulse edge is detected and a telegram is likewise sent on the bus.

The following example clarifies this:



Since only two pulse edges remain stable for the duration of the minimum operation time T_M, only these edges are recognised as valid.

5.2 Limit telegram rate

A new monitoring period starts at the end of the previous monitoring period or – in the case of bus voltage recovery – at the end of the transmission delay. The transmitted telegrams are counted. As soon as the *"Max. number of transmitted 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.

5.3 Cyclical sending

Cyclical sending is part of the "Switch sensor" function. It enables the object "Telegr. switch" to be sent automatically at fixed intervals.

If cyclical sending is only carried out for a specific object value (ON or OFF), this condition is based on the value of the communication object. It is therefore possible in principle to start the cyclical sending by sending a value to the object "Telegr. switch". Since this is generally not required, the "Write" and "Update" flags of the object are deleted in the default setting so that the object cannot be modified via the bus. If this functionality should however be required, these flags must be set accordingly.

When the object "Telegr. switch" is modified and after bus voltage recovery (once the transmission delay has elapsed), the object value is immediately sent on the bus and the transmission cycle time is restarted. The minimum value for the cyclic period is 200 ms. If a smaller value is set in the parameters, the transmission cycle time is identical to the minimum value.

5.4 Dimming

The option **"1 push button dimming"** is selected as the default setting i.e. the switching and dimming function can be fully controlled via a single push button. As a result, a telegram for dimming BRIGHTER or DARKER is sent alternately after each dimming operation. If the object "Telegr. switch" = 0, a dimming BRIGHTER telegram is sent. To be able to evaluate the status signal of the actuator, the "Write" flag of the "Telegr. switch" object must be set.

The following table clarifies the function in detail:

Value of the object “Telegr. switch”	Value of the last dimming telegram	Reaction to dimming operation (transmitted dimming telegram)
OFF	DARKER	BRIGHTER
OFF	BRIGHTER	BRIGHTER
ON	DARKER	BRIGHTER
ON	BRIGHTER	DARKER

Table 1: Dimming function “1 push button dimming”

If “2 push button dimming” is required, the function of the individual push button (e.g. “ON” or “DIM BRIGHTER”) must be set in the parameters *“Reaction on short or long operation”*. The user thus has complete freedom over

- which push buttons should be combined together in order to dim a group of luminaires
- which function the individual push button has in this case.

5.5 Scene control

Using a scene, an input controls several actuator groups at a specific, preset value via a single operation. The input can recall and/or store a scene via the function “Control scene”. A scene can be implemented here in two ways:

Scene via 5 objects

On the one hand, each input can control several actuator groups simultaneously via 5 communication objects. Several telegrams are usually sent when the scene is recalled. The storing of the scene is carried out by reading out the current values of the actuator groups via the bus. An actuator group can either be controlled via 1 bit values (ON/OFF) or 8 bit values (0…255 corresponds to 0…100%).

8 bit scene

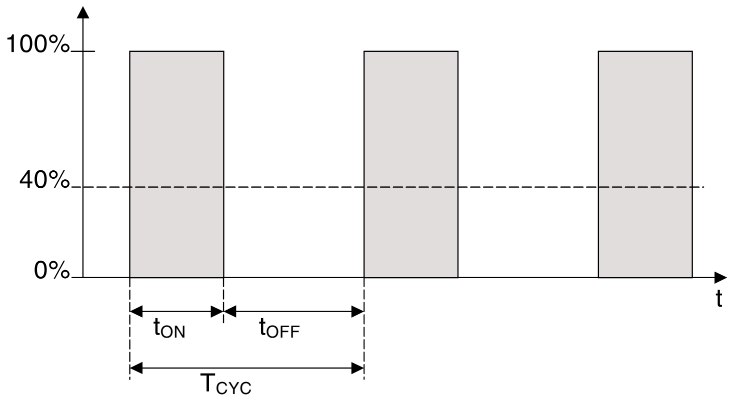
On the other hand, the input can recall values that are stored in the actuator by sending the object “8 bit scene”. This simplified scene function is only possible if it is supported by the actuator. A scene object contains a scene number (0…63) and the information as to whether the scene is recalled or stored. It is set in the actuator which scene number(s) it reacts to.

5.6 Control of electronic relay (heating actuator)

The “Heating actuator” function switches an electronic relay which is normally used to control an electrothermal valve drive. The device is usually regulated by a room thermostat. Various types of control (e.g. continuous-action control) are possible.

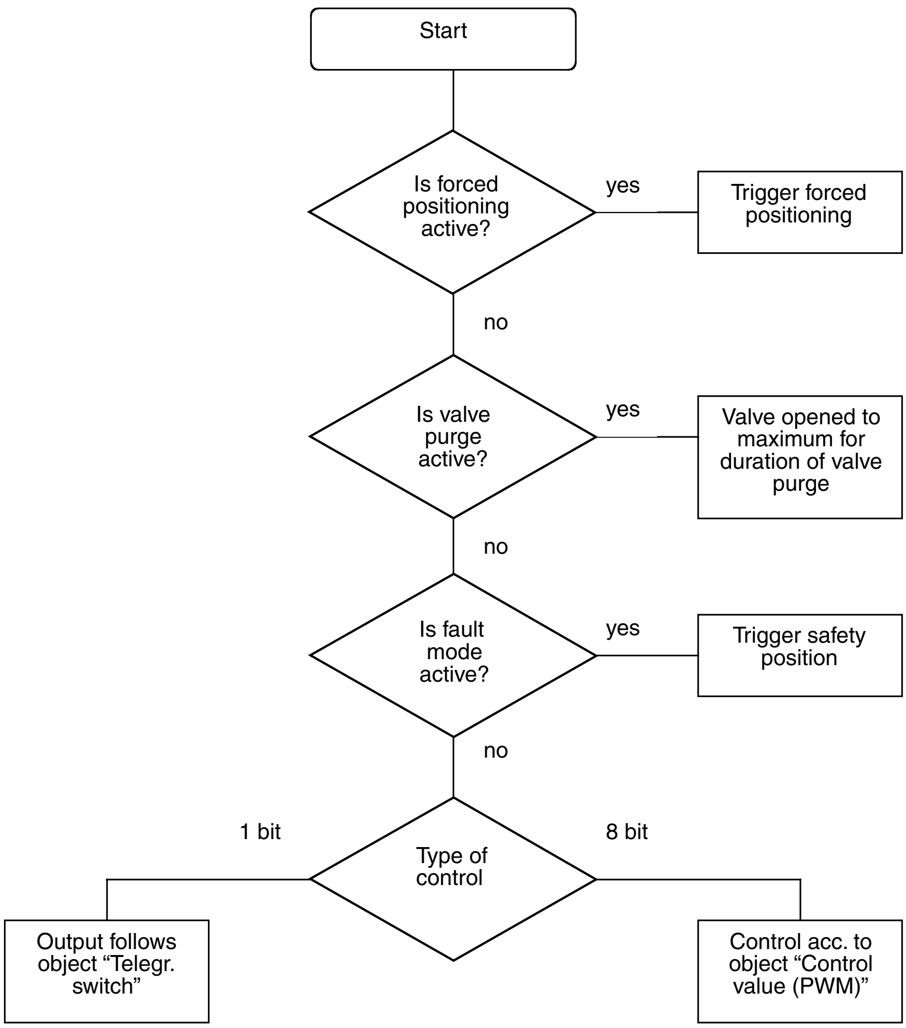
The electrothermal valve drive can be triggered via two-step control or pulse width modulation. When pulse width modulation is used, the control is carried out via a variable mark-to-space ratio.

The following example clarifies this:



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

The actuator can trigger specific special positions during “Forced positioning”, “Valve purge” and “Safety position”. The following diagram provides an overview:



To improve the control behaviour, 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:

Triggering of the valve via	Behaviour at start	Behaviour at end
Forced positioning	Trigger immediately	Once an ON or OFF phase has elapsed
Valve purge	Trigger immediately	Stop immediately
Fault mode	Once the cycle has elapsed	Once the cycle has elapsed

The sequence in the table simultaneously indicates the priority of the special positions. Forced positioning has the highest priority.

5.7 Switching sequences

The function “Switching sequence” enables up to five communication objects (1 bit) to be switched on or off via a single input.

A switching sequence consists of a succession of switching levels which represent specific object values. An operation of the input switches up or down one level.

Example: Switching sequence “sequentially on/off (one push button)” with three communication objects

Switching level		Value of the communication objects		
No.	**Short code**	**“Value 3”**	**“Value 2”**	**“Value 1”**
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>...

The defined switching sequences stand out due to the fact that only the value of a single communication object is changed between two switching levels. The following switching sequences are possible:

“sequentially on/off (one push button)”

This switching sequence switches on another communication object in succession after each operation. If all the objects are switched on, they are switched off again in succession – starting with the last object that was switched on.

“sequentially on/off (several push buttons)”

This switching sequence is similar to the function “sequentially on/off (one push button)” with the exception that it is only possible to switch up or down via an input. When the switching sequence has reached the end, further operations in the same direction are ignored. For this reason, at least two inputs are necessary for this switching sequence.

"All combinations"

This switching sequence runs through all the combinations of communication objects in succession. Only the value of one communication object is modified between two switching levels. A concrete example of this sequence is e.g. the switching of two groups of luminaires in the order
00 – 01 – 11 – 10 – 00 ...

A table of grey code can be found in the appendix under section 7.1.

Further possibilities

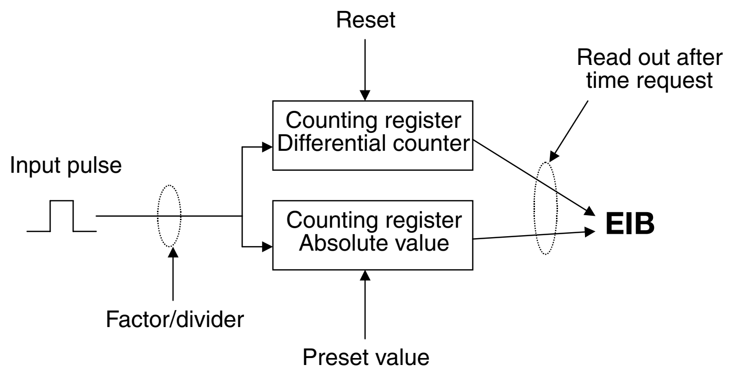
Apart from modifying the switching level via the operation of the input, it can also be changed via the communication objects "Level increment/decrement". This is used for example to switch up or down with two or more inputs. On receipt of a value at this communication object, the behaviour is identical to an operation at the input.

Note: The current switching level is continually produced from the status of the communication objects. If e.g. a communication object is modified by another device, the current switching level can also be changed as a result.

5.8 Pulse counter

The pulse counter function is used to count binary switching impulses.
It contains all the important characteristics that are required by a high-capacity counter.

The following diagram provides an overview:



Apart from the absolute counter, it is possible to enable a differential counter which measures differential values (e.g. comparable with a daily mileage indicator). To be able to adapt the count rate, factors and dividers can be set for both counters.

Both counter values can either be sent cyclically on the bus or on request. They have definable overflow values.

The overflow of the differential counter can be sent via telegram on the bus. This telegram can be used to read out the absolute counter content.
The absolute counter content is sent continually on the bus after an adjustable change.

A bus voltage failure can lead to the deletion of the counter values.

Note : The maximum counting frequency may not exceed 5 Hz.
The minimum pulse duration is 50 ms. The maximum capacitive load at the input is 22 nF.

Note: When connecting to an SO impulse output (e.g. an energy consumption meter), it must be ensured that it is electrically isolated from the mains. The correct polarity should also be observed ("-" on the black core).

5.9 Behaviour during 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 identical to "0" and the device carries out an initialisation after bus voltage recovery.

5.10 Behaviour after bus voltage recovery

This section describes the behaviour of the device after bus voltage recovery with an initialisation.

After bus voltage recovery, there is a transmission delay before telegrams are sent on the bus. The transmission delay can be set in the general parameters.

The following diagram indicates the time sequence:

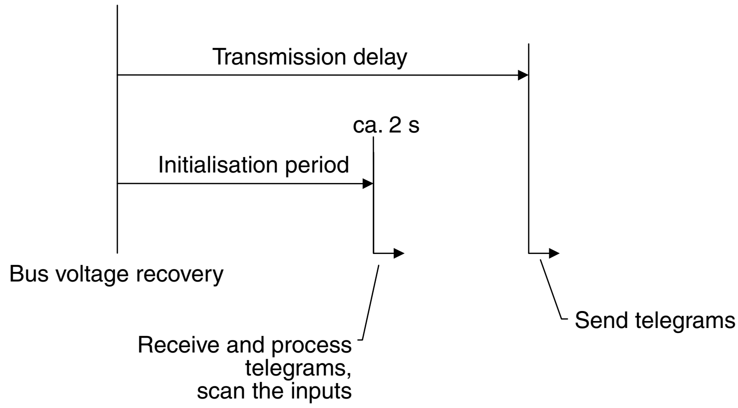


Diagram 2: Behaviour after bus voltage recovery

On bus voltage recovery, the inputs are scanned after the initialisation period and the object values are update accordingly, if possible. If the input is operated, the device behaves as if the operation had started at the end of the initialisation period.

The behaviour is dependent on the function of the channel. The following list provides an overview:

Function	Behaviour after bus voltage recovery*
Switch sensor	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. Otherwise the behaviour can be set in the parameters.
Switch/dimming sensor	No telegrams are sent on the bus.
Shutter sensor	No telegrams are sent on the bus.
Value / forced operation	Object values are overwritten with the parameterised values.
Control scene	When the scene is controlled via “5 separate objects”, the object values of the scene are overwritten with the parameterised values.
Control of electronic relay (“heating actuator”)	Until the first telegram of the room thermostat has been received, the adjustable value is set.
Control LED	The status of the output can be set in the parameters.
Switching sequence (“latching relay”)	No telegrams are sent on the bus.
Push button with multiple operation	No telegrams are sent on the bus.
Counter	No telegrams are sent on the bus.

*More precisely: Behaviour directly after the transmission delay

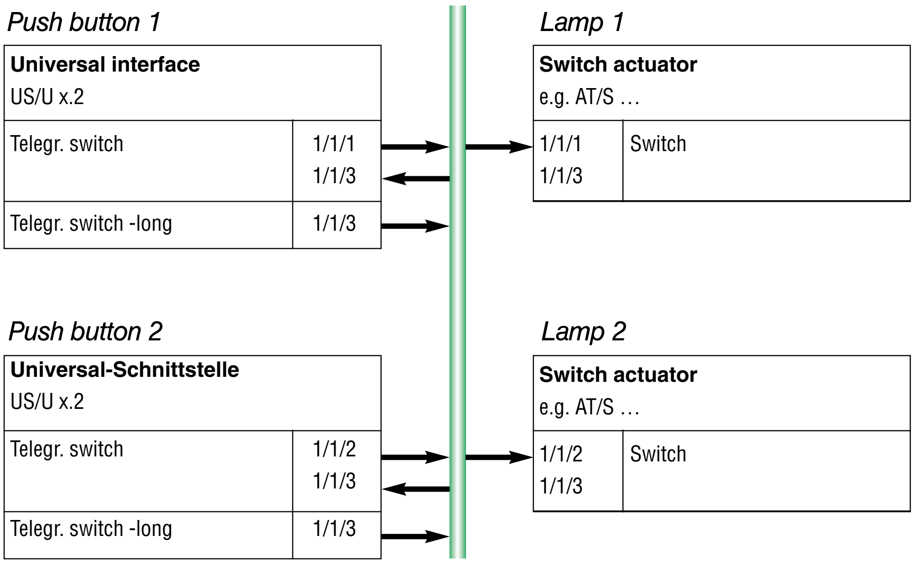
6 Planning and application

This section contains some tips and application examples for the practical use of the device.

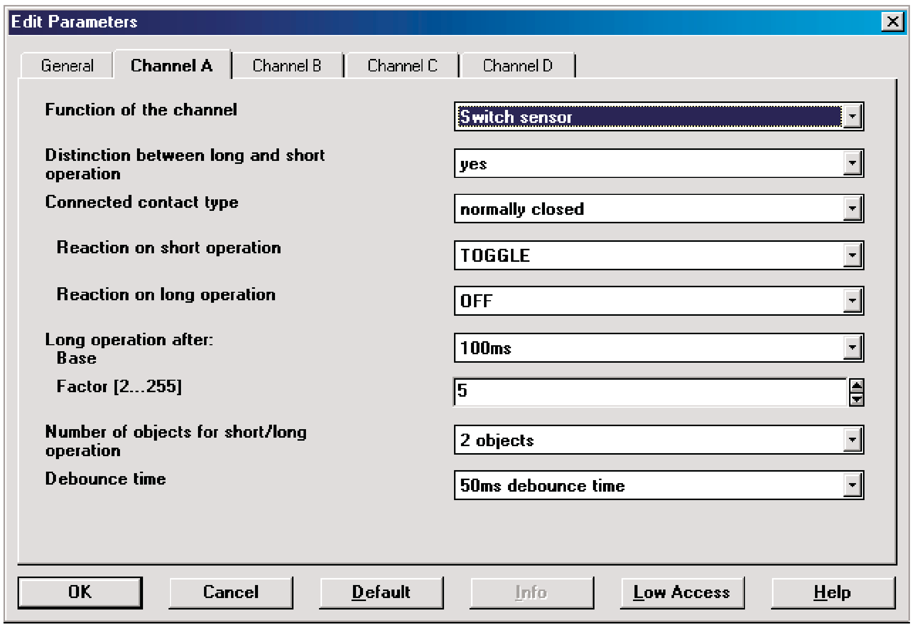
6.1 1 push button operation with central function (switch lamp)

The lighting is switched on and off after a short operation of a push button. A long operation switches the lighting off centrally.

Linking the group addresses:



Parameter settings for push button 1 and push button 2:

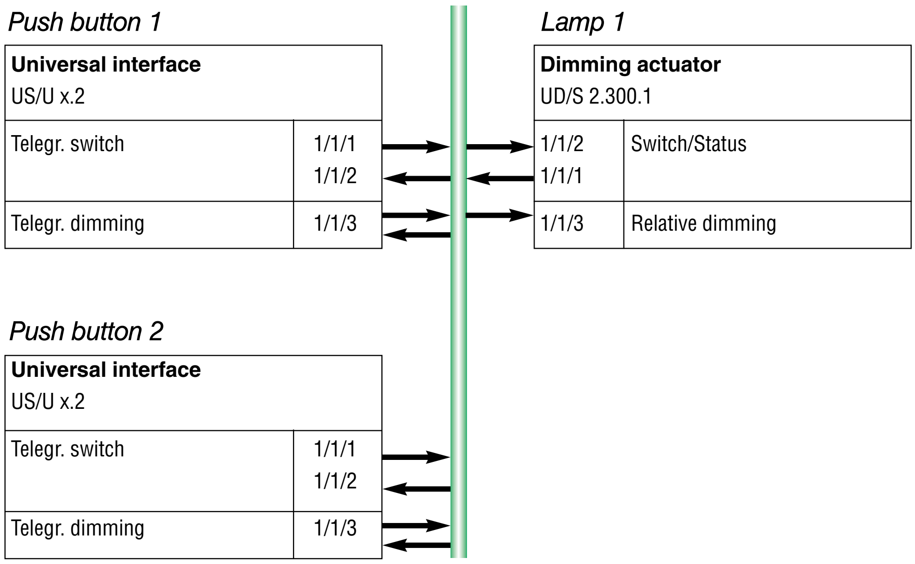


6.2 Operation of dimmable lighting

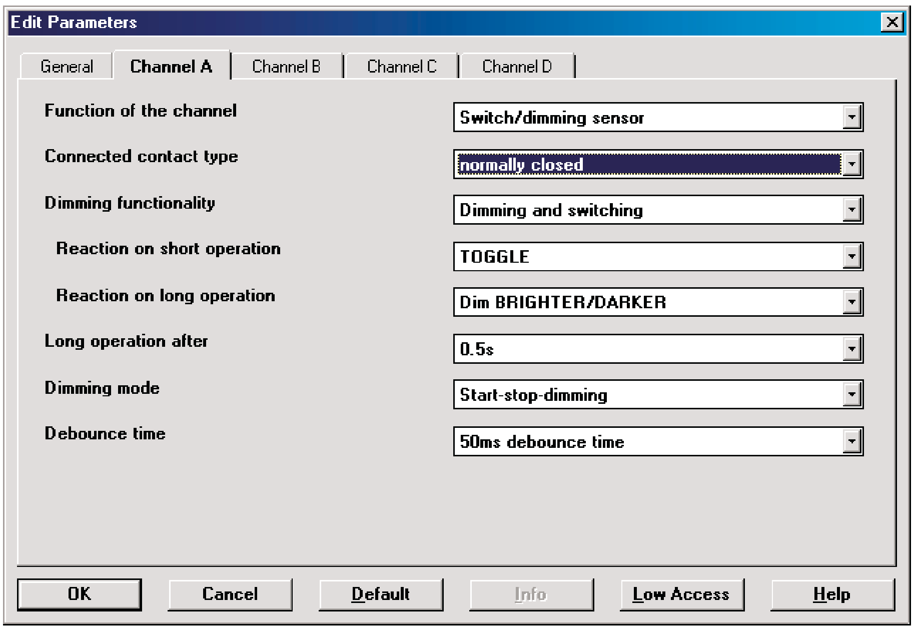
1 push button operation

A short operation switches the lighting while a long operation dims alternately brighter or darker (opposite to the last dimming process). Both push buttons operate the same lamps..

Linking the group addresses:



Parameter settings for push button 1 and push button 2:



2 push button operation

The same group address link is suitable for dimming with 2 push buttons. Modification of the parameters:

“Reaction on short operation” = “ON” or “OFF”
“Reaction on long operation” = “Dim BRIGHTER” or “Dim DARKER”

6.3 Operation of shutters

1 push button operation

Push button 1 and push button 2 operate shutter 1 from different places.
A short operation moves the shutter (opposite to last movement) while a long operation adjusts the lamella.

Linking the group addresses:

Push button 1

Universal interface US/U x.2	
Telegr. shutter UP/DOWN	1/1/1
Telegr. STOP/lamella adj.	1/1/2
Upper limit position	1/1/3
Lower limit position	1/1/4

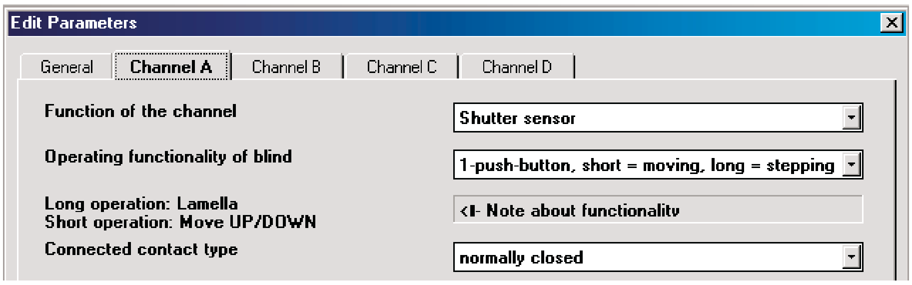
Shutter 1

Shutter actuator JA/S …	
1/1/1	Move shutter Up-Down
1/1/2	Lamella adj./Stop Up-Down
1/1/3	Telegr. status upper position*
1/1/4	Telegr. status lower position*

Push button 2

Universal interface US/U x.2	
Telegr. shutter UP/DOWN	1/1/1
Telegr. STOP/lamella adj.	1/1/2
Upper limit position	1/1/3
Lower limit position	1/1/4

Parameter settings for push button 1 and push button 2:



* The universal interfaces find out via the objects “Upper limit position” and “Lower limit position” whether the actuator is located in a limit position. This function is supported by the new generation of ABB shutter actuators (from 2003 onwards). Otherwise, 2 push button operation is recommended.

2 push button operation

Push button 1 and push button 2 operate shutter 1 from the same place.
A long operation lowers (push button 1) or raises (push button 2) the shutter.
A short operation closes (push button 1) or opens (push button 2) the lamella by a step.

Linking the group addresses:

Push button 1 (downwards)

Universal interface US/U x.2	
Telegr. shutter UP/DOWN	1/1/1
Telegr. STOPP/lamella adj.	1/1/2
Upper limit position	1/1/3
Lower limit position	1/1/4

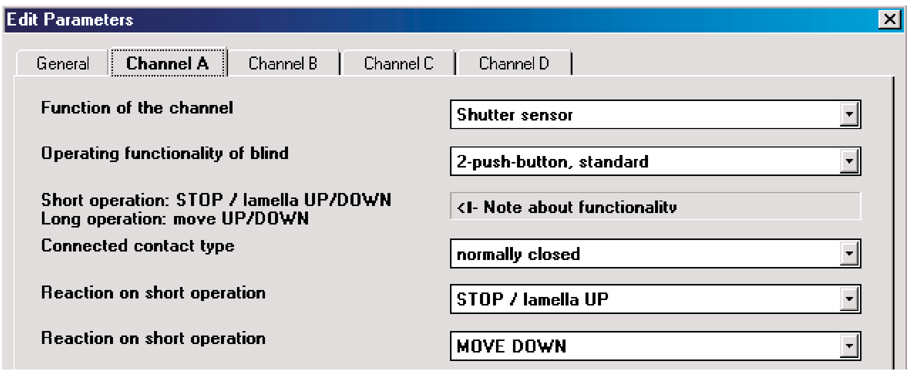
Shutter 1

Shutter actuator JA/S …	
1/1/1	Move shutter UP-Down
1/1/2	Lamella adj./Stop Up-Down
1/1/3	Telegr. status upper position*
1/1/4	Telegr. status lower position*

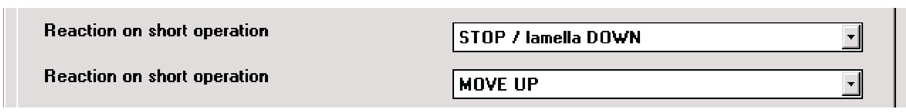
Push button 2 (upwards)

Universal interface US/U x.2	
Telegr. shutter UP/DOWN	1/1/1
Telegr. STOPP/lamella adj.	1/1/2
Upper limit position	1/1/3
Lower limit position	1/1/4

Parameter settings for push button 1:



Parameter settings for push button 2:

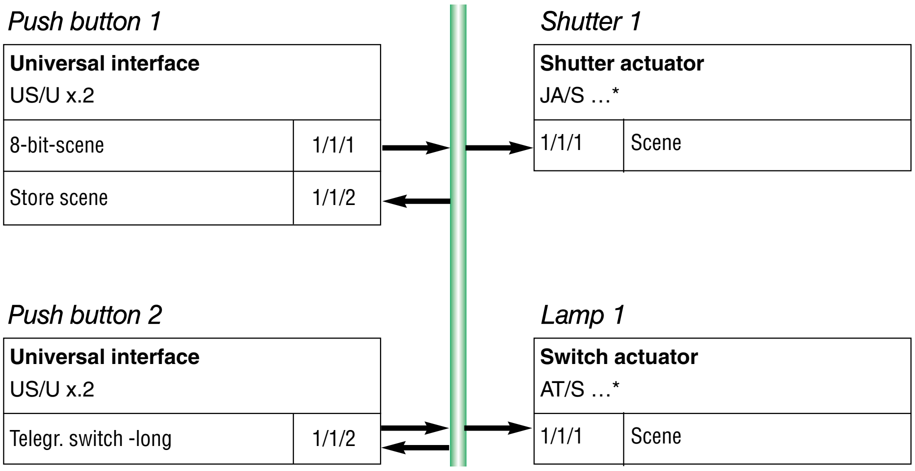


6.4 Controlling scenes

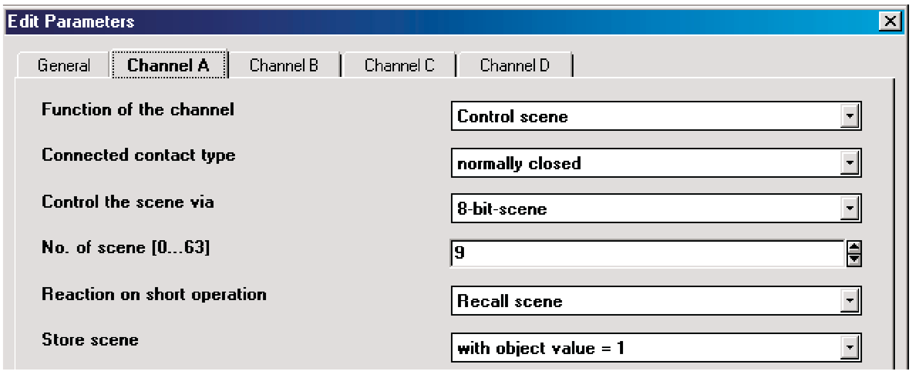
8 bit scene*

Push button 1 and push button 2 control shutter 1 and luminaire 1. Push button 1 recalls the scene. After a long operation of push button 2, the current shutter position and the status of the lighting are stored. The scene is saved in the actuator.

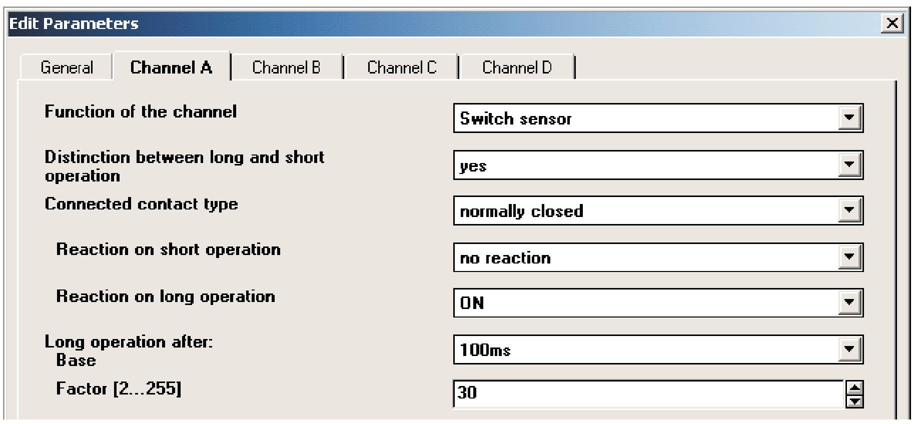
Linking the group addresses:



Parameter settings for push button 1:



Parameter settings for push button 2:

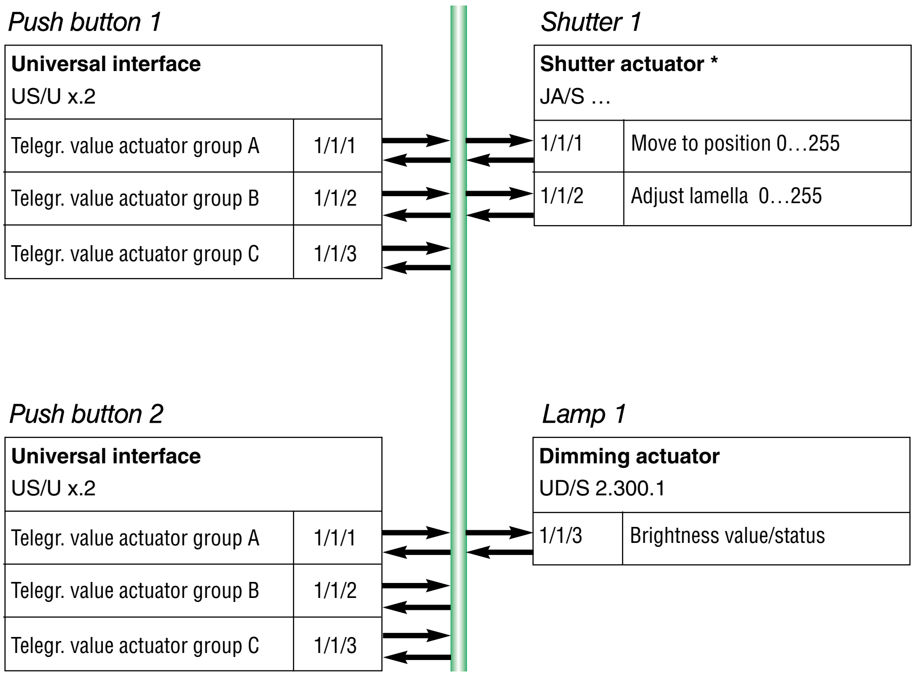


* The 8 bit scene requires actuators that support this function. This is the case with the new generation of ABB shutter actuators and switch actuators (from 2003 onwards). For other devices, it is advisable to select the option "5 separate objects".

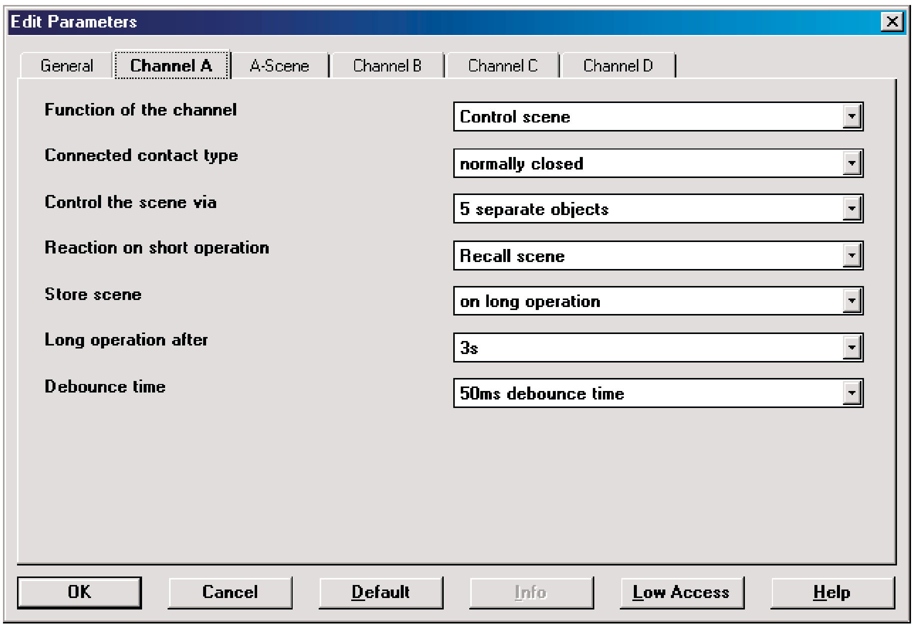
Scene control via 5 separate objects

Push button 1 and push button 2 control shutter 1 and lamp 1. A short operation recalls the scene. After a long operation, the current shutter position and the brightness value are stored. Both push buttons store different scene values.

Linking the group addresses:



Parameter settings for push button 1 and push button 2:

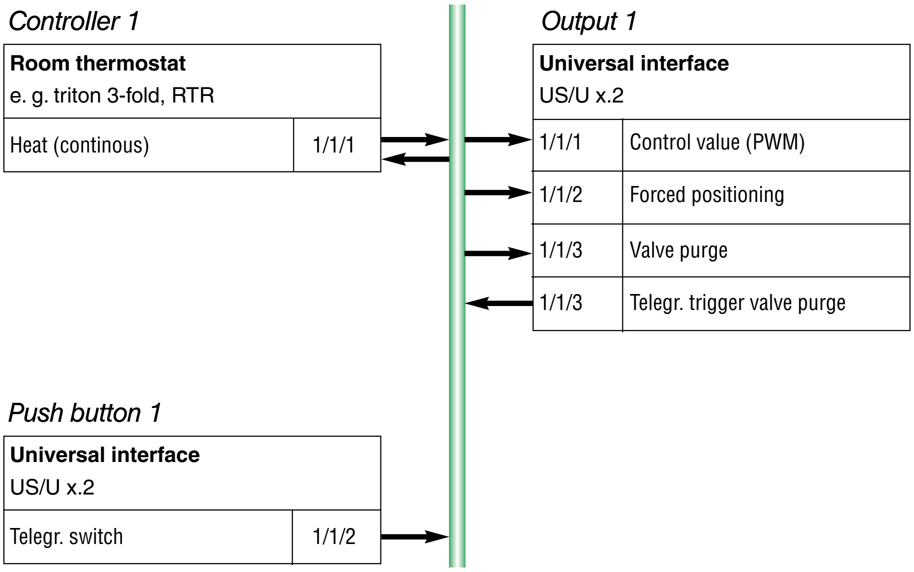


* This function is only available for shutter actuators that can move into position via an 8 bit value.

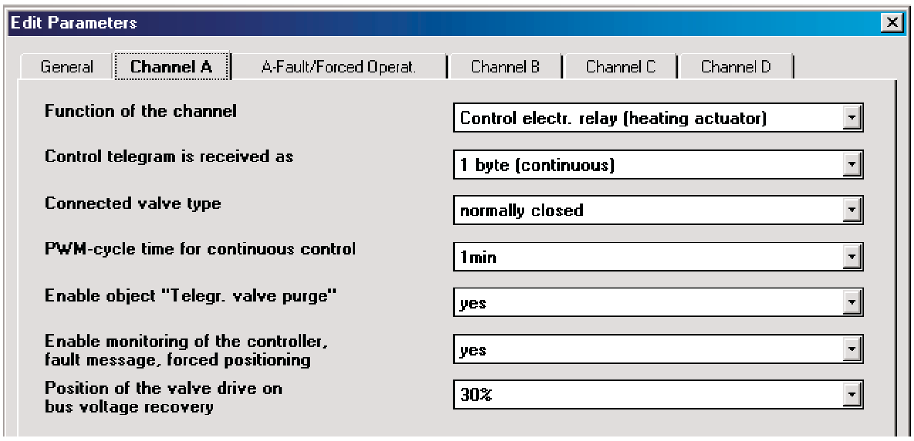
6.5 Controlling a heating valve

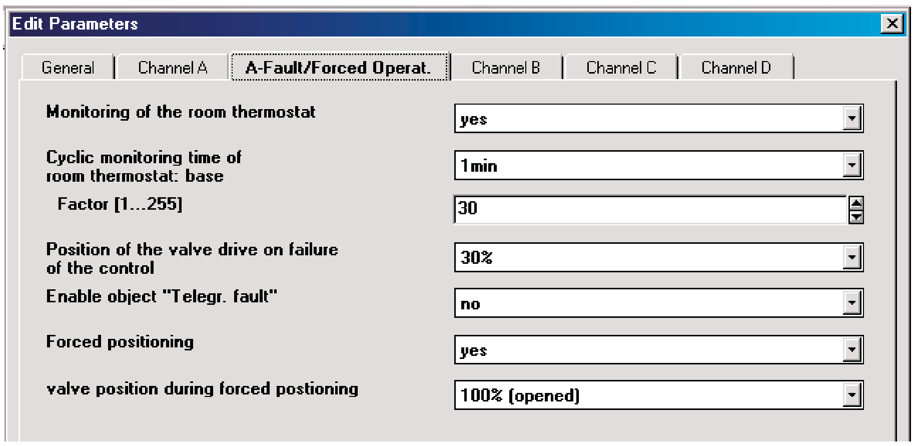
An electronic relay ER/U 1.1, which controls an electrothermal valve drive, is connected to output 1 of a universal interface. The room temperature is regulated continuously via controller 1. The valve is purged once a week, whereby it is opened for approx. 5 minutes. The valve can be fully opened via a forced operation of push button 1. If no telegrams have been received from controller 1 within 30 minutes, the valve is opened by 30% (fault mode).

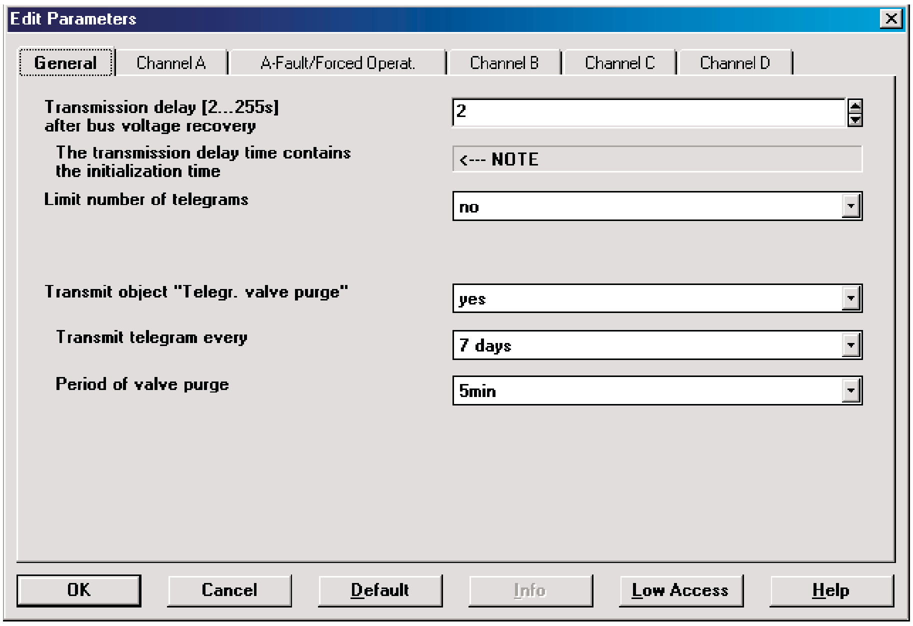
Linking the group addresses:



Parameter settings for output 1:







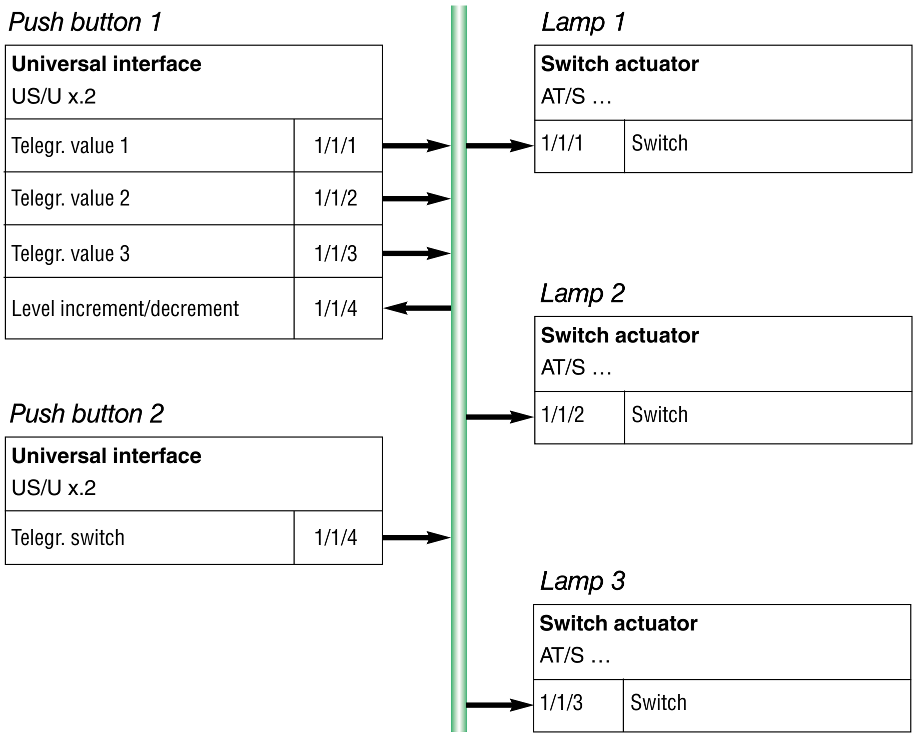
Output 1 sends itself the group address 1/1/3 once a week and thereby triggers the valve purge. The sending object is enabled in the “General” parameter window.

6.6 Switching the lighting in sequence

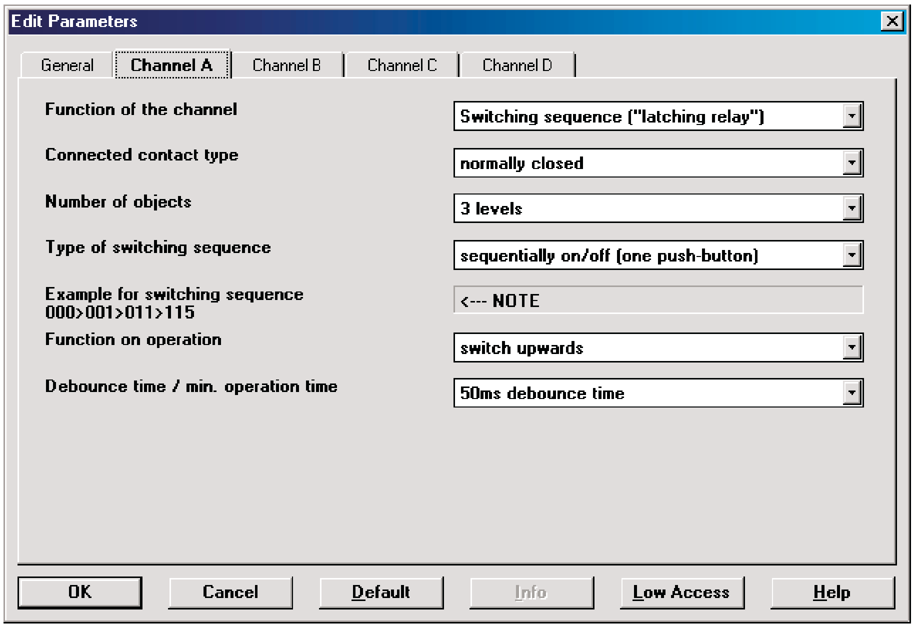
Sequentially on/off (several push buttons)

Push button 1 and push button 2 control a luminaire with three independent circuits – lamp 1, lamp 2, lamp 3. Push button 1 switches them on in sequence (order: lamp 1>lamp 2>lamp 3). Push button 2 switches them off in sequence (order: lamp 3>lamp 2>lamp 1).

Linking the group addresses:



Parameter settings for push button 1:



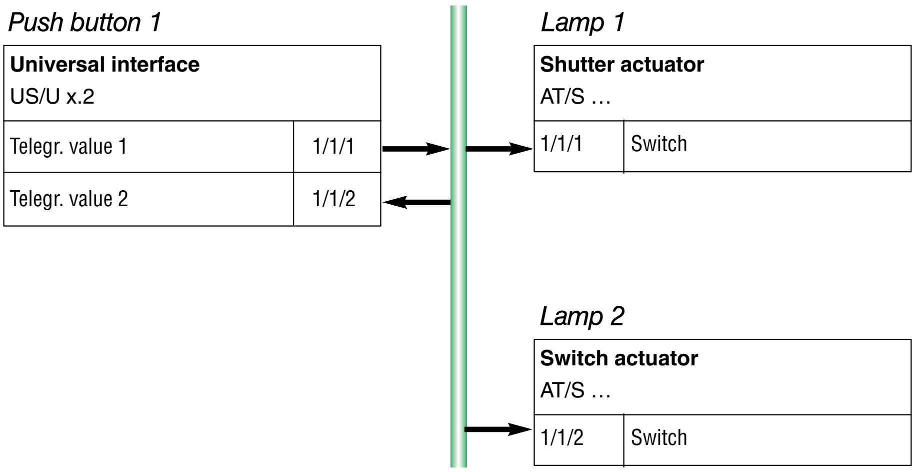
The parameters of push button 2 should be set so that “Telegr. switch” sends a “0” with each push button action.

All combinations

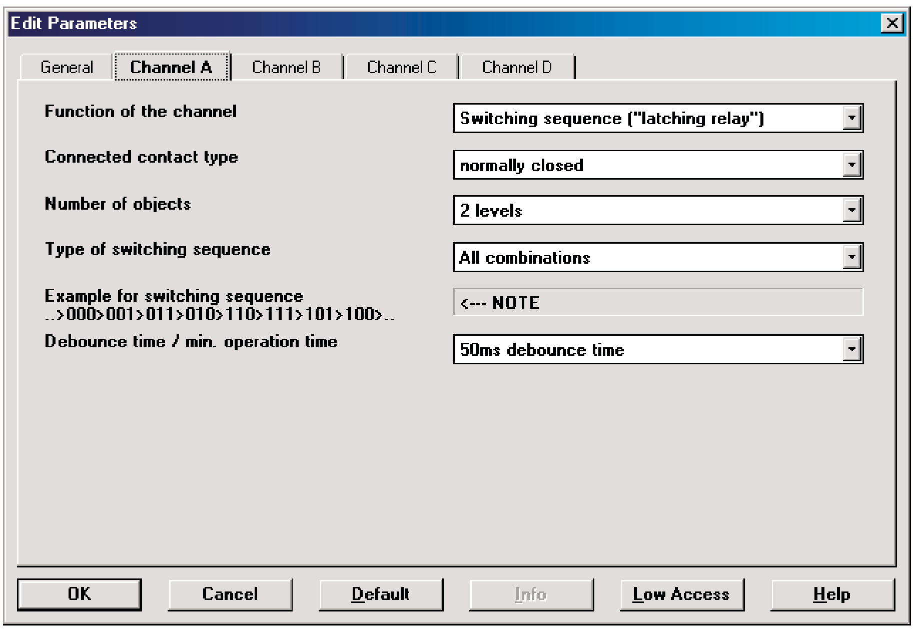
Push button 1 controls a luminaire with two independent circuits – lamp 1 and lamp 2. After operation, it switches through all the possible combinations in the following order:

	Lamp 1	*Lamp 2*
Output state	OFF	OFF
1st operation	ON	OFF
2nd operation	ON	ON
3rd operation	OFF	ON
4th operation	OFF	OFF
etc.		

Linking the group addresses:



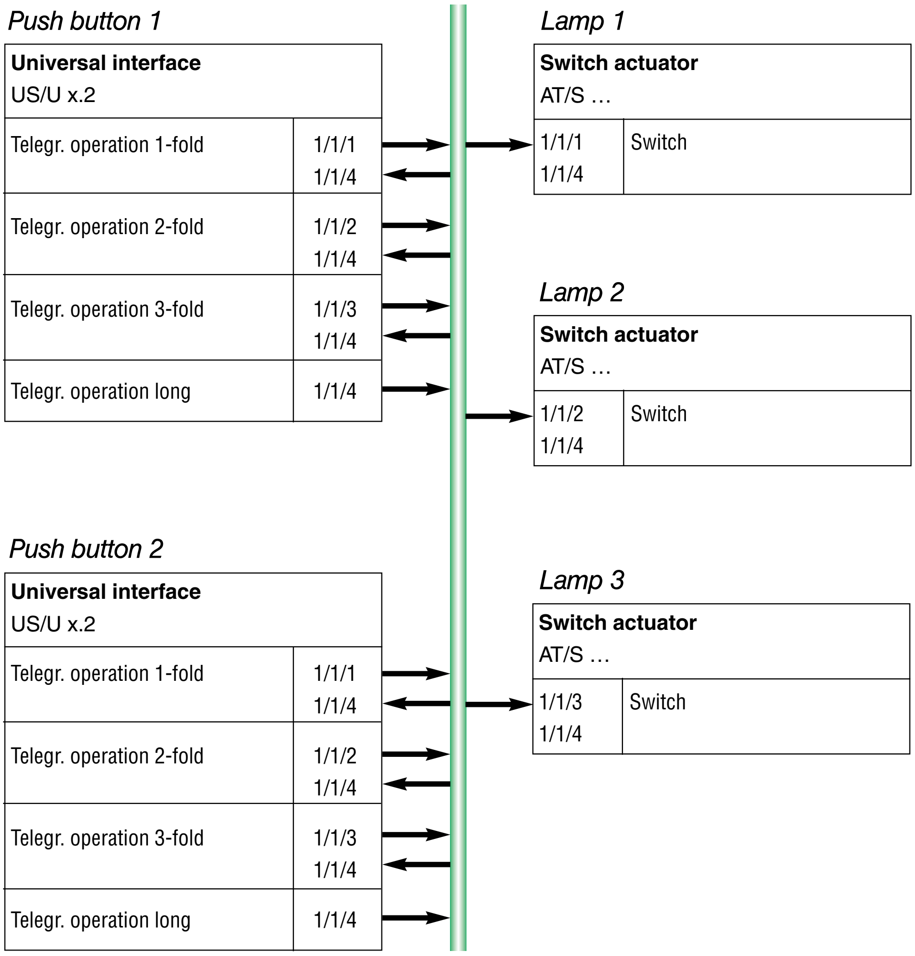
Parameter settings for push button 1:



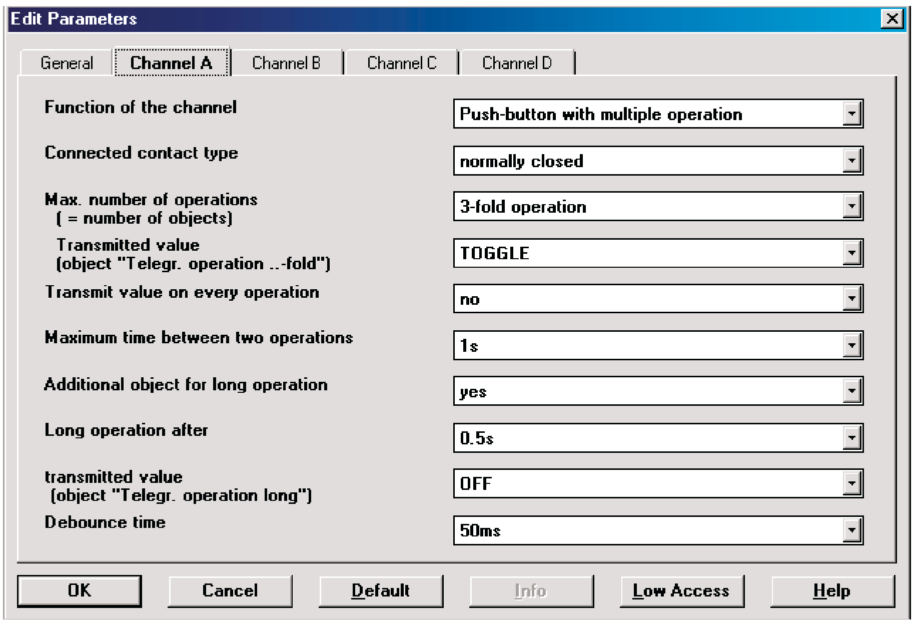
6.7 Switching the lighting via multiple push button operation

Push button 1 and push button 2 control lamp 1, lamp 2 and lamp 3. After one push button action, lamp 1 is toggled. Lamp 2 is toggled after a two-fold operation while lamp 3 is toggled after a three-fold operation. Lamp 1, lamp 2 and lamp 3 are switched off after a long push button action.

Linking the group addresses:



Parameter settings for push button 1 and push button 2:



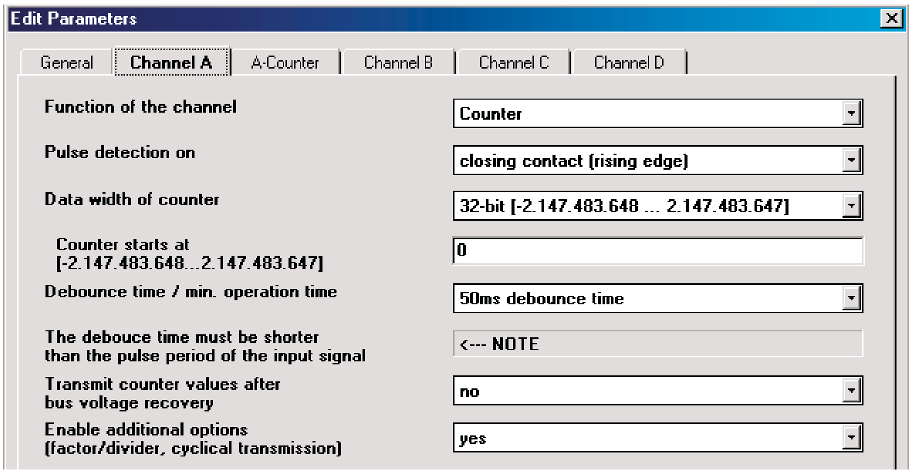
6.8 Counting energy values

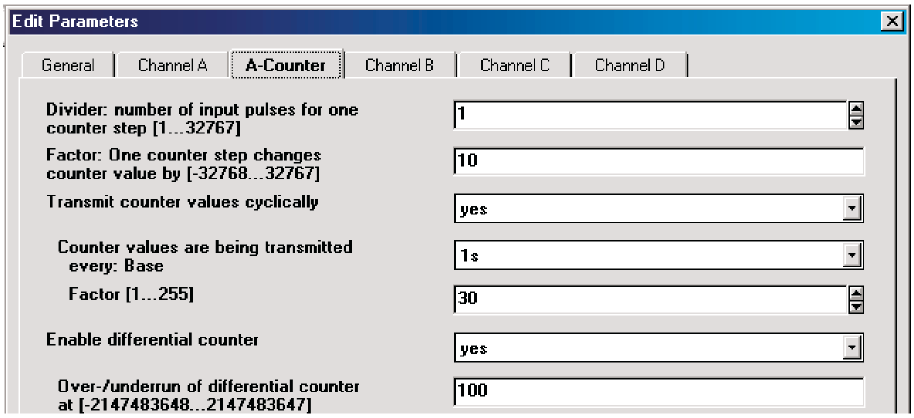
Input 1 is connected to the S0 output of an energy consumption meter (100 pulses/kWh). The 4 byte counter value is displayed on the bus as Wh. It is sent on the bus every 30 seconds and if the value changes by 100 Wh.

Input 1

Universal interface US/U x.2	
Telegr. counter value 4 bytes	1/1/1 →
Request counter values	1/1/2 ←
Differential counter overflow	1/1/2 →

Parameter settings for input 1:





100 pulses per kWh means 1 pulse per 10 Wh. The counter is therefore increased every impulse by the value 10 (factor = 10).

The differential counter generates an overflow every 100 Wh, which in turn requests a new counter value.

7 Appendix

7.1 Table of grey code

The sequence is characterised by the fact that only one value changes between two levels. The transition to the next level therefore only requires the sending of a single telegram.

The following table describes the grey code when 5 objects are used:

Switching level		**Value of the communication objects**				
No.	**Short code**	**“Value5”**	**“Value4”**	**“Value3”**	**“Value2”**	**“Value1”**
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

7.2 Ordering information

Description	Ordering info. Short code	Product code	bbn 40 16779 EAN	Unit price [EURO]	Price group	Weight 1 Pc. [kg]	Package [Pc.]
Universal interface, 2-fold	**US/U 2.2**	GH Q631 0074 R0111	56483 0		26	0.05	1
Universal interface, 4-fold	**US/U 4.2**	GH Q631 0070 R0111	56481 6		26	0.05	1

Notizen

The information in this leaflet is subject to change without further notice.

Pub No. 2CDC 504 019 D0201

Your EIB-Partner