

**01514**

2 inputs/outputs interface for led KNX

---

### **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.

# Contents

<b>1. General</b>	4
1.1 Product and functional overview	4
<b>2. Device technology</b>	5
2.1 Technical data	5
2.2 Device connection	6
2.3 Description of the inputs and outputs	6
2.4 Assembly and installation	6
<b>3. Commissioning</b>	7
3.1 Overview	7
3.2 General functions	8
3.3 Function "Switch sensor"	10
3.4 Function "Switch/dimming sensor"	13
3.5 Function "Shutter sensor"	15
3.6 Function "Value / forced operation"	18
3.7 Function "Control scene"	21
3.8 Function "Control electr. relay (heating actuator)"	24
3.9 Function "Control LED"	28
3.10 Function "Switching sequence"	31
3.11 Function "Push-button with multiple operation"	33
3.12 Function "Counter"	35
3.13 Programming	37
<b>4. Special functions</b>	38
4.1 Debounce time and minimum operation time	38
4.2 Limit rate of telegrams	38
4.3 Cyclic sending	38
4.4 Dimming	39
4.5 Control scene	39
4.6 Control electr. relay ("heating actuator")	39
4.7 Switching sequence	41
4.8 Count pulses	43
4.9 Reaction on bus voltage failure	43
4.10 Reaction after bus voltage recovery	43
<b>5. Planning and application</b>	44
5.1 1 button operation with central function (Switch light)	44
5.2 Operation of dimmable illumination	45
5.3 Operation of blinds	46
5.4 Control of scenes	47
5.5 Control of a heater valve	48
5.6 Switching of lighting in switching sequence	50
5.7 Switching of lighting via multiple button pushes	51
5.8 Metering energy values	52
<b>6. Appendix</b>	54
6.1 Gray code table	54

## General

---

### 1. General

The comprehensive functions provided in modern buildings with bus KNX must be simple and intuitive for the user to operate. At the same time, clarity and comfortable operation are highly significant factors with regard to the sense of perceived value of a building installation.

The Universal Interfaces fulfil the individual demands both in the functional building as well as in the private sector. In the same manner, the planners of systems are presented with a whole range of application options regarding the realization of functions.

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 Interfaces 01514 serve as the interface for comfortable operation of systems via conventional push buttons/switches or for reading out technical binary signals. Furthermore, they facilitate the control of LEDs as well as the Electronic Relay for controlling electro-thermal control valves.

The very compact design allows the installation of conventional 60 mm installation box, e.g. behind a conventional push button or switch.

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
- Counting of impulses and operations
- Reading out of technical contacts (e.g. relays)

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

## Device technology

### 2. Device technology

In this section, the device functions of the Universal Interfaces are explained. The devices feature two channels, which can be parameterized as either an input or output with the software ETS.

Using the colour-coded connecting cables, it is possible to connect conventional push buttons, floating contacts, LEDs or Electronic Relays can be connected. Series resistors for operation of the LEDs are also integrated into the device. The contact scanning voltage and the supply voltage for the LEDs are made available by the device.

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

#### 2.1 Technical data

<b>Power supply:</b>	Bus voltage	Current consumption < approx. 10 mA
	Number	2 can be separately parameterized as inputs or outputs
<b>Inputs/outputs:</b>	Permissible cable length	≤10 m
	Sensing voltage	20 V DC (pulsed)
<b>Input:</b>	Input current	0,5 mA
	Output voltage	3...5 V DC
<b>Output:</b>	Output current	Max. 2 mA, limited by 1.5 kΩ series resistors
	Safety	Short-circuit protected, overload protected, reverse voltage polarity protected
<b>Operating and display elements:</b>	LED (red) and button	For assignment of the physical address
<b>Connections:</b>	Inputs/outputs	4 cables approx. 30 cm long, can be extended to max. 10 m
	Bus KNX	Via bus connection terminals, included in scope of delivery
<b>Ambient temperature range:</b>	Operation	- 5°C ... 45°C
	Storage	25° C ... 55° C
	Transport	-25° C ... 70° C
<b>Other:</b>	Enclosure	IP 20 (EN 60529) in the installed state
	Safety class	III
	CE mark	in accordance with the EMC guideline and low voltage guideline
	Approval	KNX certified
	Installation	In switch box Ø 60 mm
	Dimensions (WxHxD):	39 x 40 x 12 mm
	Weight	0,05 kg

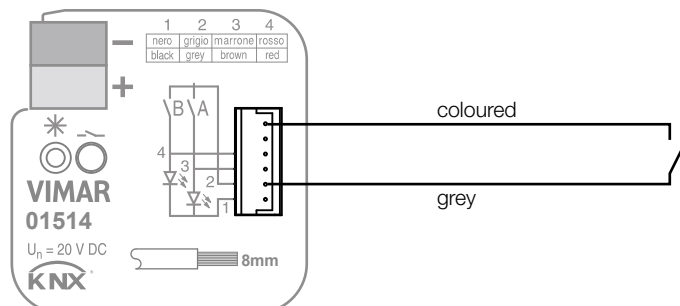
## Device technology

### 2.2 Device connection

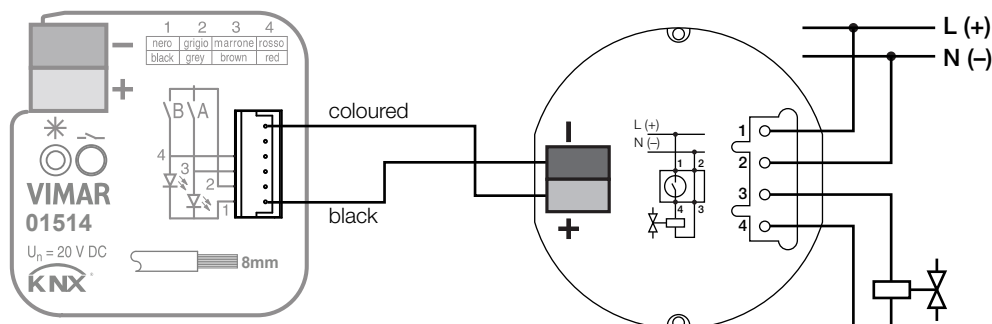
When operated as an input, the contact to be scanned is connected between the grey and the coloured core.

When operated as an output, the load (LED or Electronic Relay) is connected between the black or coloured core. The coloured core provides the positive output voltage.

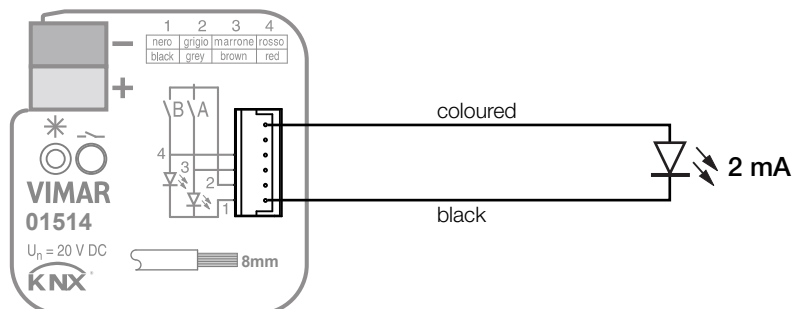
**Connection of a floating push button/switch:**



**Connection of an Electronic Relay ER/U 1.1**



**Connection of a LED**



### 2.3 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
red	Channel B

**Black core: 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.

### 2.4 Assembly and installation

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

## Commissioning

### 3. Commissioning

#### 3.1 Overview

The Universal Interface features a high-performance application program *"Binary Input Display Heat, 2f/1.x"*. Programming requires the ETS software.

Application program	Number of communication objects	Max. number of group addresses	Max. number of associations
Binary Input Display Heat, 2f/1.x	15	254	255

The following functions can be set separately for every 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 via and Electronic Relay ERU 1.1. 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 ("impulse relay")	For operation of several actuator groups in the predefined sequence.
Push-button with multiple operation	For triggering of different functions depending on the frequency of operation Even a long operation can be detected and a function triggered.
Counter	For counting input impulses Different meter data types can be set. An additional differential meter enables metering, e.g. of daily values. Factor/dividers facilitate different metering speeds.

#### 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. After import in the ETS, it is available in the ETS  
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.

## Commissioning

### 3.2 General functions

In this section, the parameters and objects that apply for the entire device are explained.

Parameters and objects that are assigned to each channel are described in the following section using output A as an example.

#### 3.2.1 General parameters

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

<b>General</b>	Transmission delay [2...255s] after bus voltage recovery	2
Channel A	The transmission delay time contains the initialization time	<--- NOTE
Channel B	Limit number of telegrams	yes
Channel C	Max. Number of transmitted telegrams within a period	20
Channel D	Period	10s
	Transmit object "Telegr. valve purge"	yes
	Transmit telegram every	30 days
	Period of valve purge	2 min

Sending delay after bus voltage recovery in s [2...255]	2...255
Limit number of telegrams	yes no
Max. Number of transmitted telegrams within period of observation	0...20...255
Period	50 ms...500 ms...1 s...10 s/30 s / 1 min
Transmit object "Telegr. valve purge"	yes no
Transmit telegram every	7 / 14 / 30 / 50 days
Period of valve purge	1 / 2 / 5 / 10 min

#### Sending delay after bus voltage recovery

The sending 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 objects are read via the bus during the transmission delay (e.g. from visualisation terminals), these requests are stored and are answered, if necessary, once the sending delay has elapsed.

For a detailed description of the reaction on bus voltage recovery see section 4.10.

#### 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 transmitted telegrams within a period of observation") can be sent within an adjustable period ("Period").

Details concerning the method of function of the telegram rate limitation can be found in section 4.2.

#### Parameter "Transmit object "Telegr. valve purge"

This function is only relevant if one or more channels are 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.



### 3.2.2 General communication objects

#### Object "Block": 1 bit

This object is visible for every channel operated as an 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 blocking of an input during operation, the reaction 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..

Telegram value	"0": enable input
	"1": block input

#### Object "Telegr. trigger valve purge": 1 bit

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

The object is set at regular intervals for an adjustable "Period of valve purge" to the value "1" 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.

## Commissioning

### 3.3 Function "Switch sensor"

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

#### 3.3.1 Parameters

Parameter window without distinction between long and short operation:

General	Function of the channel	Switch sensor
Channel A	Distinction between long and short operation	no
Channel B	Cyclic transmission of object "Telegr. switch"	always
Channel C	Reaction on closing the contact (rising edge)	ON
Channel D	Reaction on opening the contact (falling edge)	OFF
	Telegram is repeated every ("transmission cycle time"): base	1s
	Factor [1...255]	30
	Transmit object value after bus voltage recovery	no
	Debounce time / min. operation time	50ms debounce time

Operating mode of the channel	Switch sensor
Distinction between long and short operation	yes no
Cyclic transmission of object "Telegr. 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 ("transmission cycle time"): base	1 s / 10 s / 1 min / 10 min / 1 h
Factor [2...255]	1...30...255
Send object value after bus voltage recovery	yes no
Debounce time / min. operation time	10 ms...50 ms...150 ms debounce time Minimum operation time

Parameter window with distinction between long and short operation

## Commissioning

General	Function of the channel	Switch sensor
Channel A	Distinction between long and short operation	yes
Channel B	Connected contact type	normally closed
Channel C	Reaction on short operation	ON
Channel D	Reaction on long operation	OFF
	Long operation after: Base	100ms
	Factor [2...255]	5
	Number of objects for short/long operation	1 object
	Debounce time	50ms debounce time

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: 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 object 2 objects
Debounce time	10 ms...50 ms...150 ms debounce time

### Parameter "Distinction between long and short operation"

In this parameter, you set if the input differentiates between short and long operation. With option "yes", after opening/closing of the contact it must first of all be ascertained if a short or long operation has occurred here. Only thereafter will a possible reaction be triggered.

The following table shows the function in detail:

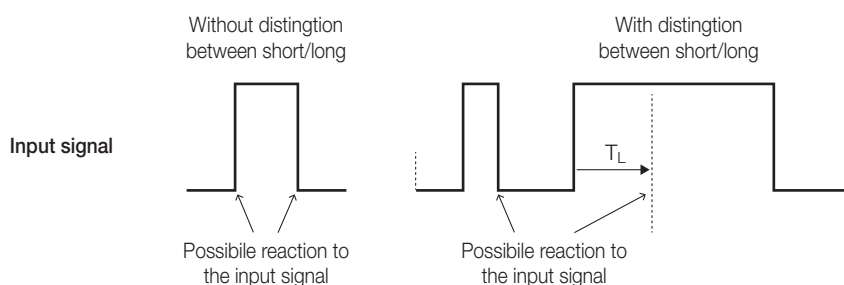


Fig. 1: Distinction between short/long operation for function "Switch"

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

## Commissioning

---

### Parameter “Cyclic transmission of object “Telegr. switch”

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

The communication object “Switch” can be sent cyclically, e.g. for lifesign monitoring of the sensor.

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

Should the parameter value “if telegram switch = ON” or “if telegram switch = OFF” be set, the corresponding object value is sent cyclically.

More detailed information on the topic of “Cyclic sending” can be found in section 4.3.

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

This parameter is visible if there is no distinction between short and long operation. For each edge you can set if the object value is to be switched ON, OFF or TOGGLE, or if no reaction should occur.

If cyclical sending has been parameterized, it is possible by setting the parameter value “terminate cyclic sending” with an operation of the input, to stop cyclic sending without a new object value being sent.

### Parameter “Telegram is repeated every (“transmission cycle time”)

This parameter is visible if cyclical transmission is active.

The send cycle time describes the time used between two cyclically transmitted telegrams:

Sending cycle time = Time base x Factor

### Parameter “Connected contact type”

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

Here you set if the contact on the input is a normally closed contact or normally open contact.

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

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

It can be set for every operation (short or long) on an input, how the object value can be changed. The object value is updated as soon as it has been determined if a short or long operation has occurred.

### Parameter “Long operation after”

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

$T_L = \text{Time base} \times \text{Factor}$

### Parameter “Number of objects for short/long operation”

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

In order to differentiate between long and short operation, by setting the parameter value “2 objects”, a further object can be enabled that solely responds to a long operation.

### Parameter “Send object value after bus voltage recovery”

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

It can be set whether the current status of the input is sent on the bus (via object “Telegr. 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 generally sent on the bus after bus voltage recovery.

### Parameter “Debounce time / min. operation 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. A minimum operation time can only be set if there is no distinction between short and long operation.

## 3.3.2 Communication objects

### Object “Telegr. switch”: 1 bit

In accordance with the parameter setting, this communication object can be switched by operation of the ON, OFF or TOGGLE input.

### Object “Telegr. switch long”: 1 bit

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 objects”. This additional object is assigned to the long operation.

## Commissioning

### 3.4 Function “Switch/dimming sensor”

In the following, all parameters and objects are described, which are visible, when the input is operated with the function “Switch/dimming sensor”. The function facilitates the operation of dimmable lighting. 1 button operation is possible. Further details about the dimming functionality can be found in section 4.4.

#### 3.4.1 Parameters

General	Function of the channel	Switch/dimming sensor
Channel A	Connected contact type	normally open
Channel B	Dimming functionality	Dimming and switching
Channel C	Reaction on short operation	TOGGLE
Channel D	Reaction on long operation	Dim BRIGHTER/DARKER
	Dimming direction after switching ON	DARKER
	Long operation after	0.5s
	Dimming mode	Start-stop-dimming
	Debounce time	50ms debounce time

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

#### Parameter “Connected contact type”

Here you set if the contact on the input is a normally closed contact or normally open contact.

#### Parameter “Dimming functionality”

This parameter is used to define if the lighting can only be dimmed (“Only dimming”) or if additional switching is also permitted (“Dimming and switching”). In this case, a long button push dims and a short button push switches.

The advantage of the “Only dimming” function is that no distinction is made between short and long operation. The dim command is issued immediately after operation; it is not necessary to wait for a confirmation of a long operation.

#### Parameter “Reaction on short operation”

This parameter is visible if in the parameter “Dimming functionality” the value “Switch and Dimming” is set. A short operation changes the value of the object “Telegr. switch”.

This parameter sets if the object “Telegr. switch” toggles with short operation (typically: 1 button dimming) or only switches OFF or ON (typically: 2 button dimming).

## Commissioning

---

### Parameter “Reaction on long operation”

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

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

### Parameter “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”).

### Parameter “Long operation after”

This parameter is visible if in the parameter “*Dimming functionality*” the value “Switch and Dimming” is set. Here the time period TL after which an operation is considered a “long” operation is defined.

### Parameter “Reaction on operation”

This parameter is visible if “Only dimming” dimming functionality is set. A distinction is not made between short and long operation here. The meaning of the adjustable values corresponds with the parameter “*Reaction on long operation*” (see above).

### Parameter “Dimming functionality”

Normal “Start-stop-dimming” starts the dimming process with a telegram BRIGHTER or DARKER and ends the dimming process with a STOP telegram. Cyclic sending of the telegram is not necessary in this case.

With “Dimming steps”, the dimming telegram is sent cyclically during a long operation. The STOP telegram ends the dimming process at the end of operation.

### Parameter “Brightness change on every sent telegram”

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

### Parameter “Transmission 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.

### Parameter “Debounce time / min. operation 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. A minimum operation time can only be set if in parameter “*Dimming functionality*” the function “Only dimming” has been set.

## 3.4.2 Communication objects

### Object “Telegr. switch”: 1 bit

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.

### Object “Telegr. dimming”: 4 bit

A long operation of 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 operation.

## Commissioning

### 3.5 Function "Shutter sensor"

In the following, all parameters and objects are described, which are visible, when the input is operated with the function "Shutter sensor". The function allows the operation of shutters and blinds with buttons or switches. 1 button operation and 1 switch operation is possible.

#### 3.5.1 Parameters

General	Function of the channel	Shutter sensor
Channel A	Operating functionality of blind	2-push-button, standard
Channel B	Short operation: STOP / lamella UP/DOWN Long operation: move UP/DOWN	<-- Note about functionality
Channel C	Connected contact type	normally closed
Channel D	Reaction on short operation	STOP / lamella DOWN
	Reaction on short operation	MOVE DOWN
	Long operation after	0.5s
	Debounce time	30ms

Operating mode of the channel	Shutter sensor
Operating functionality of 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
Telegram "Lamella" is repeated every ...	0,3 s...0,4 s...10 s
Debounce time	10 ms...30 ms...150 ms debounce time

#### Parameter "Operating functionality"

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 *

**\* Note:** If an actuator is in a limit position (see objects "Upper limit position" or "Lower limit position"), the direction of motion is predefined.

In 1 push button/switch operation, 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 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 adjustment. UP" → ... or ... → "Move DOWN" → "STOP/Lamella adjustment DOWN" → ...

2-push-button, stepping	
On operation	"Stop/Lamella adjustment UP" or "... DOWN"

### Parameter "Connected contact type"

Here you set if the contact on the input is a normally closed contact or normally open contact.

### Parameter "Reaction on operation"

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

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

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

### Parameter "Long operation after"

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

### Parameter "Telegram "Lamella" is repeated every ..."

This parameter is visible in operations in which the object "Telegr. STOP / lamella adj." is sent cyclically on the bus during long operation. Here the time between two telegrams is set.

### Parameter "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.



## Commissioning

---

### 3.5.2 Communication objects

#### Object "Telegr. shutter UP/DOWN": 1 bit

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.

Telegram value	"0"	UP
	"1"	DOWN

#### Object "Telegr. STOP / lamella adj.": 1 bit

This communication object sends a STOP command or lamella adjustment.

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

#### Object "Upper limit position": 1 bit

Using this object, the shutter actuator indicates if it is in the upper limit position ("shutter/blind open"). The object is intended for 1-button operation.

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

#### Object "Lower limit position": 1 bit

Using this object, the shutter actuator indicates if it is in the lower limit position ("shutter/blind closed"). The object is intended for 1-button operation.

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

## Commissioning

### 3.6 Function “Value / forced operation”

In the following, all parameters and objects are described, which are visible, when the input is operated with the function “Value / forced operation”. The function allows the sending of values of any data types.

#### 3.6.1 Parameters

Parameter window without distinction between long and short operation:

General	Function of the channel	Value / forced operation
Channel A	Connected contact type	normally closed
Channel B	Distinction between long and short operation	no
Channel C	Reaction on operation	1-byte-value [0...255]
Channel D	Transmitted value [0...255]	0
	Transmit object value after bus voltage recovery	no
	Debounce time / min. operation time	50ms debounce time

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 position) 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]
Transmitted value	Dependent on the selection made at <i>Reaction on operation</i>
Send object value after bus voltage recovery	yes no
Debounce time / min. signal time	10 ms...50 ms...150 ms debounce time Minimum signal time

## Commissioning

Parameter window with distinction between long and short operation

General	Function of the channel	Value / forced operation
Channel A	Connected contact type	normally closed
Channel B	Distinction between long and short operation	yes
Channel C	Reaction on short operation	1-byte-value [0...255]
Channel D	Transmitted value [0...255]	0
	Reaction on long operation	1-byte-value [0...255]
	Transmitted value [0...255]	0
	Long operation after: Base	100ms
	Factor [1...255]	4
	Debounce time	50ms debounce time

Reaction on short operation	no reaction 1-bit-value 2-bit-value (forced position) 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]
Transmitted value	Dependent on the selection made at <i>Reaction on operation</i>
Reaction on long operation	no reaction 1-bit-value 2-bit-value (forced position) Valore 1 byte [0...255] 2-byte-value [-32768...32767] 2-byte-value [0...65535] 2-byte-value (floating point) 4-byte-value [0...4294967295]
Transmitted value	Dependent on the selection made at <i>Reaction on operation</i>
Long operation after: 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

### Parameter "Connected contact type"

This parameter determines whether the contact on the input is "closed" when operated (normally opened contact) or "opened" (normally closed).

### Parameter "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.

### Parameter "Reaction on operation"

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

## Commissioning

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

This parameter is visible if there is a distinction between short and long operation. It determines the data type that is sent with a long or short operation.

### Parameter “Transmitted value”

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

### Parameter “Long operation after”

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

$T_L = \text{Time base} \times \text{Factor}$

### Parameter “Send object value after bus voltage recovery”

This parameter is visible if there is no distinction between short and long operation. If “yes” is set here, the device sends the object “Value” on the bus after bus voltage recovery (once the sending delay has elapsed).

### Parameter “Debounce time / min. operation 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. A minimum operation time can only be set if there is no distinction between short and long operation.

### 3.6.2 Communication objects

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

Data width, type	Value range	EIS type	Typical application
1 bit	0, 1	EIS 1	Switch command
2 bit	0, 2, 3	EIS 8	Forced positioning
1 byte Unsigned	0...255	EIS 6	Brightness value, position value
2 byte, integer, with sign	-32768...+32767	EIS 10	Counter value
2 byte, integer, Unsigned	0...65535	EIS 10	Counter value
2 byte, floating point value	-100...+100	EIS 5	Temperature values
4 byte, integer, Unsigned	0...4294967295	EIS 11	Counter value

\*sends values with fixed exponent of 3

### Object “Telegr. value (...)” (various data types)

This communication object sends a value on the bus when opening or closing of the contact. The value and data type can be freely set in the parameters.

With a distinction between long and short operation 2 objects are visible for every input. One object only transmits during short operation, the other object only during a long operation.

**Note:** By default, the value objects of the “Write” flag (exception with 1 bit objects) are deleted. 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 parameterized value after bus voltage recovery.

## Commissioning

### 3.7 Function "Control scene"

In the following, all parameters and objects are described, which are visible, when the input is operated with the function "Control scene". This function facilitates the recall and storing of states of several actuator groups. A detailed explanation of the function can be found in section 4.5.

#### 3.7.1 Parameters

Parameter window with control of the scene via "5 separate objects":

General	Function of the channel	Control scene
Channel A	Connected contact type	normally closed
A-Scene	Control the scene via	5 separate objects
Channel B	Reaction on short operation	Recall scene
Channel C	Store scene	no
Channel D	Debounce time	50ms debounce time

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

Parameter window with control of the scene via "8 bit scene":

General	Function of the channel	Control scene
Channel A	Connected contact type	normally closed
Channel B	Control the scene via	8-bit-scene
Channel C	No. of scene	Scene no. 1
Channel D	Reaction on short operation	Recall scene
	Store scene	no
	Debounce time	50ms debounce time

No. of scene	Scene no. 1 ... Scene no. 64
--------------	------------------------------------

## Commissioning

Additional parameter window "A-Scene" with control of the scene via "5 separate objects":

General	Control of actuator group A via	1-bit-object
Channel A	Preset value actuator group A	ON
<b>A-Scene</b>	Control of actuator group B via	1-bit-object
Channel B	Preset value actuator group B	ON
Channel C	Control of actuator group C via	1-bit-object
Channel D	Preset value actuator group C	ON
	Control of actuator group D via	1-bit-object
	Preset value actuator group D	ON
	Control of actuator group E via	1-bit-object
	Preset value actuator group E	ON

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

### Parameter "Connected contact type"

This parameter determines, whether the contact on the input is "closed" (normally opened contact) or "opened" (normally closed) when operated.

### Parameter "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" (see section 4.5 for further information).

### Parameter "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.

### Parameter "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. The following table provides an overview:

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

Parameter value	Reaction
"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 "Switch/Telegr. value actuator group A...E" are read via the bus and stored in the object values. During the ongoing long operation, the objects "Switch/Telegr. value actuator group A...E" can be modified via the bus. After the end of a long operation, the object "Store scene" sends the value "0" on the bus and the current object values are stored in the device.
"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. A long as the object value is "1", the objects "Switch/Telegr. value actuator group A...E" can be modified via the bus. On receipt of the object value "0", the current object values are then stored in the device. <b>Important: Storage of the current scene thus requires the object values "1" and "0" to be sent in succession.</b>
"on long operation AND object value = 1"	If the object "Store scene" receives the value "1" on the bus, the next long button operation leads to sending the value "1" via the object "Store scene". As a result, the object values "Switch/Telegr. value actuator group A...E" are scanned via the bus. After the end of a long operation, the object values "Switch/Value actuator group" is stored in the device. 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. The same applies for the case where the object "Store scene" has last received the value "0".

## Commissioning

Control the scene via "8 bit scene"

Parameter value	Reaction
"on long operation"	On a long button operation, the object "8 bit scene" sends a store 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" receives the value "1" on the bus. 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. The same applies for the case where the value "0" has been last received.

### Parameter "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 operation is considered a "long" operation is set.

### Parameter "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.

### Tab "A-Scene"

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

### Parameter "Control of actuator group A...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 "Switch/Value Actuator group A to ...E" is set accordingly.

### Parameter "Preset value actuator group A...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 and renewed recall of the scene, the current object values of the actuator groups A...E are overwritten with the values set here.

## 3.7.2 Communication objects

### Object: "Telegr. Switch actuator group A...E": 1 bit or "Telegr. value actuator group A...E": 8 bit

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 parameterized). When storing the scene, the device reads out the current value via the bus and stores it in these objects.

### 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
	(MSB) (LSB)
M:	0 – Scene is recalled 1 – Scene is stored
x:	Not used
S:	No. of scene (0...63)

### Object "Store scene": 1 bit

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 functions depends on the type of scene storage.

More detailed information is available under the description of the parameter "Store scene".

## Commissioning

### 3.8 Function "Control electr. relay (heating actuator)"

In the following, all parameters and objects are described, which are visible, when the input is operated with the function "Control electr. relay". A thermal valve drive for heating valves can be operated via an electronic relay. Further detailed explanations of the function can be found in section 4.6.

#### 3.8.1 Parameters

General	Function of the channel	Control electr. relay (heating actuator)
Channel A	Control telegram is received as	1 bit (PWM or on-off-control)
Channel B	Connected valve type	normally closed
Channel C	PWM-cycle time for continuous control	1min
Channel D	Enable object "Telegr. valve purge"	no
	Enable monitoring of the controller, fault message, forced positioning	no
	Position of the valve drive on bus voltage recovery	20%

Operating mode of the channel	Control electronic relay (heating actuator)
Control telegram is received as	1 bit (PWM or on-off-control) 1 byte (continuous)
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 "Telegr. valve purge"	yes <u>no</u>
"Enable monitoring of the controller, fault message, forced positioning"	yes <u>no</u>
Position of the valve drive on bus voltage recovery	0% (closed) ... <u>20 %</u> ... 100% (opened)



## Commissioning

Additional parameter window at "Enable monitoring of the controller, fault message, forced positioning" = "yes":

General	Monitoring of the room thermostat	yes
Channel A	Cyclic monitoring time of room thermostat: base	1min
<b>A-Fault/Forced Operat.</b>	Factor [1...255]	20
Channel B	Position of the valve drive on failure of the control	10%
Channel C	Enable object "Telegr. fault"	no
Channel D	Forced positioning	no

Monitoring of the room thermostat	yes no
Cyclic monitoring time of room thermostat: base	1 s / 10 s / <u>1 min</u> / 10 min / 1 h
Factor [1...255]	1...20...255
Position of the valve drive on failure of the control	0 % (closed) ... <u>10 %</u> ... 100% (opened))
Enable object "Telegr. fault"	yes no
Forced positioning	yes no
Valve position during forced positioning	0 % (closed) ... <u>50 %</u> ... 100% (opened))

### Parameter "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 controls the heating actuator via normal switching commands. A 2-step control or pulse width modulation of the control value can be easily 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 (see graphic in section 4.6).

### Parameter: "Connected valve type"

In this parameter, you can set whether a valve should be controlled as "normally closed" or "normally open". If the valve is normally closed, the opening of the valve is achieved via the closing of the electronic relay. If the valve is normally open, the opening of the valve is achieved by the opening of the relay.

### Parameter: "PWM cycle time for continuous control"

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

With 1 bit control and 1 byte control, this time is only used during control of the actuator in fault mode, during forced positioning and directly after bus voltage recovery.

### Parameter "Enable object "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.

## Commissioning

### Parameter: "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 programmed value is used as the PWM cycle time.

### Parameter window "A-Fault/Forced Operat."

This parameter window is visible if the value "yes" has been entered in the parameter "Enable cyclic monitoring time of room thermostat, fault messages, forced positioning" in the parameter window "A: Function".

### Parameter "Monitoring of the room thermostat"

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

The telegrams of the room thermostat are transferred to the electronic actuator 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 there are no telegrams for the cyclic monitoring time to the objects "Switch" or "Control value (PWM)" during the period defined in this parameter, the actuator switches to fault mode and triggers a safety position. The fault mode is ended as soon as a telegram is received.

### Parameter "Cyclic monitoring time of room thermostat"

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

Duration = Time base x Factor.

### Parameter "Position of the valve drive on failure of the control"

This parameter defines the safety position which 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"

With this parameter, the object "Telegr. fault" can be enabled. 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.

### Parameter "Forced positioning"

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

### Parameter "valve position during forced positioning"

The valve position triggered by the actuator during the forced operation is defined in this parameter. The PWM cycle time  $T_{CYC}$  of the control is defined in the parameter "cycle time for continuous control".

## 3.8.2 Communication objects

### Object "Switch": 1 bit

This object is visible if the control of the heating actuator is implemented 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

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.

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

## Commissioning

---

### Object "Valve purge": 1 bit

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" End valve purge  
                          "1" Start valve purge

### Object "Forced operation": 1 bit

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.

Telegram value:       "0" End forced positioning  
                          "1" Start forced positioning

### Object "Telegr. status/ackn.": 1 bit

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 will be opened

**Note:** For PWM continuous control, this object is sent after each change in the output. For this reason, the additional telegram load should be considered particularly with short PWM cycle times.

### Enable object "Telegr. fault": 1 bit

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.

Telegram value:       "0" no fault  
                          "1" Fault mode active

## Commissioning

### 3.9 Function "Control LED"

In the following, all parameters and objects are described, which are visible, when the input is operated with the function "Control LED". An operation can be acknowledged, for example, via an LED.

#### 3.9.1 Parameters

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

General	Function of the channel	Control LED
Channel A	LED functionality	switch ON/OFF
Channel B	LED is switched ON, if	Object "Telegr. switch" = 1
Channel C	Time limit of LED-control	yes
Channel D	Time limit: base	10s
	Time limit: factor [1...255]	5
	Transmit status via object "Telegr. status/ackn."	no
	State of LED on bus voltage recovery	OFF

Operating mode of the channel	Control LED
LED functionality	switch ON/OFF Flashing
LED is switched ON, if	Object "Telegr. switch" = ON Object "Telegr. switch" = OFF
Time limit of LED-control	yes no
Time limit: base	1 s / 10 s / 1 min / 10 min / 1 h
Time limit: factor [1...255]	1...5...255
Transmit status via object "Telegr. status/ackn."	yes no
State of LED on bus voltage recovery	OFF ON

## Commissioning

Parameter window for "LED function" = "Flashing":

<b>General</b>		
<b>Channel A</b>	Function of the channel	Control LED
Channel B	LED functionality	Flashing
Channel C	LED flashes, if	Object "LED flashing" = 1
Channel D	LED is switched ON for	1s
	LED is switched OFF for	1s
	Time limit of LED-control	yes
	Time limit: base	10s
	Time limit: factor [1...255]	5
	Transmit status via object "Telegr. status/ackn."	yes
	State of LED on bus voltage recovery	OFF

LED flashes, if	Object "LED flashing" = ON Object "LED flashing" = OFF
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 limit of LED-control	yes no
Time limit: base	1 s / 10 s / 1 min / 10 min / 1 h
Time limit: factor [1...255]	1...5...255
Transmit status via object "Telegr. status/ackn."	yes no
State of LED on bus voltage recovery	OFF ON

### Parameter "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.

### 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 in which state of the object "LED switch" the LED is switched on.

### Parameter "LED flashes, if"

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

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

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

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.

### Parameter "Time limit of LED-control"

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

## Commissioning

---

### 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.

Duration = Time base x Factor

### Parameter "Transmit status via object, "Telegr. status/ackn."

The object "Telegr. status/ackn." is enabled with this object. The object value is ON if the LED has been switched on or flashes.

## 3.9.2 Communication objects

### Object "LED switching": 1 bit

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.

### Object "LED flashing" = 1 bit

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.

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

### Object "LED permanent ON": 1 bit

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.

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

### Object "Telegr. status/ackn.": 1 bit

This object is visible if the parameter "*Send status via ...*" the value "yes" is set. The state of the output is fed back.

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

## Commissioning

### 3.10 Function “Switching sequence”

In the following, all parameters are described, which are visible, when the input is operated with the function “Switching sequence”. A switching sequence facilitated step-by-step modification of several values with just a single operation.

Next switching step = actual value of the object  $\pm 1$

+1 → switch upwards

-1 → switch downwards

#### 3.10.1 Parameters

General	Function of the channel	Switching sequence ("latching relay")
Channel A	Connected contact type	normally closed
Channel B	Number of objects	3 levels
Channel C	Type of switching sequence	sequentially on/off (one push-button)
Channel D	Example for switching sequence	<--- NOTE ...>000>001>011>111>011>001>000>...
	Debounce time / min. operation time	50ms debounce time

Operating mode of the channel	Switching sequence ("latching relay")
Connected contact type	normally closed <u>normally open</u>
Number of objects	2 / <u>3</u> / 4 / 5 levels
Type of switching sequence	<u>sequentially on/off (one push-button)</u> sequentially on/off (multiple push-buttons) Switch all combinations (Gray code)
Function on operation	<u>switch upwards</u> switch downwards
Debounce time / min. signal time	10 ms... <u>50 ms</u> ... 150 ms debounce time Minimum signal time

#### Parameter “Connected contact type”

This parameter determines, whether the contact on the input is “closed” (normally opened contact) or “opened” (normally closed) when operated.

#### Parameter “Number of operations”

The number of steps (max. 5) is the same as the number of communication objects. The objects “Value 1” to “Value n” are enabled.

#### Parameter “Type of switching sequence”

The switching sequence can be selected here. Every sequence has different object values for each switching sequence. The following switching sequences are possible (a detailed description can be found in section 4.7):

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

The example relates to the state of three objects (“0” = OFF, “1” = ON). A table with the Gray code can be found in section 6.1.

## Commissioning

---

### Parameter “Function on operation”

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

### Parameter “Debounce time / min. operation 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.

### 3.10.2 Communication objects

#### Objects “Value 1” to “Value 5”: 1 bit

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

#### Object “Level increment/decrement”: 1 bit

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.

Telegram value:	"0"	Switch downwards
	"1"	Switch downwards



## Commissioning

### 3.11 Function “Push-button with multiple operation”

In the following, all parameters and objects are described, which are visible, when the input is operated with the function “Multiple operation”.

If the output is operated multiple times within a determined time, a defined object value can be changed regardless of the number of operations. In this way, for example, different light scenes are possible by multiple button operations.

#### 3.11.1 Parameters

General	Function of the channel	Push-button with multiple operation
Channel A	Connected contact type	normally closed
Channel B	Max. number of operations ( = number of objects)	3-fold operation
Channel C	Transmitted value (object "Telegr. operation ...fold")	TOGGLE
Channel D	Transmit value on every operation	no
	Maximum time between two operations	1s
	Additional object for long operation	yes
	Long operation after	0.5s
	transmitted value (object "Telegr. operation long")	TOGGLE
	Debounce time	50ms

Operating mode of the channel	Push-button with multiple operation
Connected contact type	normally closed normally open
Max. number of operations (Number of objects)	1-fold operation 2-fold operation 3-fold operation 4-fold operation
Transmitted value (object "Telegr. operation ...fold")	ON OFF TOGGLE
Transmit value on every operation	si 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
transmitted value (object "Telegr. operation long")	ON OFF TOGGLE
Debounce time	10 ms...50 ms...150 ms debounce time

#### Parameter “Connected contact type”

This parameter determines, whether the contact on the input is “closed” (normally opened contact) or “opened” (normally closed) when operated.

#### Parameter “Max. number of operations”

The maximum number of possible operations is set here. The number is equal to the number of communication objects “...fold operation”. 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.

## Commissioning

---

### Parameter “Transmitted 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”.

### Parameter “Transmit 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.

### Parameter “Maximum time between two operations”

The time that can elapse between two operations is set here. After an operation, the device will wait for the time entered here. If there are no further operations within this period, the object “Switch” is sent and counting will restart with the next detected operation.

### Parameter “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.

### Parameter “Long operation after”

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

### Parameter “Transmitted value”

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

### Parameter “Debounce time / min. operation 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. A minimum operation time can only be set if there is no distinction between short and long operation.

## 3.11.2 Communication objects

### Objects “1-fold operation” to “4-fold operation”: 1 bit

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.

### Object “Long operation”: 1 bit

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.

## Commissioning

### 3.12 Function “Counter”

In the following, all parameters and objects are described, which are visible, when the input is operated with the function “Counter”.

With the function “Counter”, the device can count the number of edges on the input. For this purpose, an “Intermediate meter” is available in addition to the normal “Meter”. Both are controlled via the number of pulses, but otherwise count independently of one another. The meters always have the same data band width as the intermediate meters.

#### 3.12.1 Parameters

General	Function of the channel	Counter
Channel A	Pulse detection on	closing contact (rising edge)
Channel B	Data width of counter	32-bit [-2.147.483.648 ... 2.147.483.647]
Channel C	Counter starts at	0
Channel D	Debounce time / min. operation time	50ms debounce time
	The debounce time must be shorter than the pulse period of the input signal	<--- NOTE
	Transmit counter values after bus voltage recovery	no
	Enable additional options (factor/divider, cyclical transmission)	no

Operating mode of the channel	Counter
Pulse detection on	closing contact (rising edge) opening contact (falling edge)
Data width of counter	8-bit [0...255] 16-bit [-32,768...32,767] 16-bit [0...65,535] 32-bit [-2,147,483,648...2,147,483,647]
Counter starts at	Dependent on the selection made at <i>Data width of counter</i>
Debounce time	10 ms... <u>50</u> ms...150 ms debounce time
Transmit counter values after bus voltage recovery	yes <u>no</u>
Enable additional options (factor/divider, cyclical transmission)	yes <u>no</u>

Additional parameter window with “*Enable additional options (...)*” = “yes”:

General	Divider: number of input pulses for one counter step [1...32767]	1
Channel A	Factor: One counter step changes counter value by [-32768...32767]	1
A-Counter	Transmit counter values cyclically	no
Channel B	Enable differential counter	no
Channel C		
Channel D		

Divider: number of input pulses for one counter step [1...32767]	<u>1</u> ...32767
Factor: One counter step changes counter value by [-32768...1...32767]	-32768... <u>1</u> ...32767
Transmit counter values cyclically	yes <u>no</u>
Counter values are being transmitted every: Base	<u>1</u> s / 10 s / 1 min / 10 min / 1 h
Factor (1...255)	1... <u>30</u> ...255
Enable differential counter	yes <u>no</u>
Over-/underrun of differential counter at [1...255]	1... <u>10</u> ...255

### Parameter “Pulse detection on”

In this parameter, the type of input signal is determined. You can determine whether the contact is a normally closed contact (opens) or a normally opened contact (closes).

### Parameter “Data type of meter reading”

In this parameter the data type of the counter (absolute counter and intermediate counter) is determined. The data type determines the counter range in which it is possible to meter.

The type of objects “Meter reading ...” and “Differential counter ...” are adapted to the data width.

### Parameter “Counter starts at”

In this parameter, the start value of the absolute counter is determined. The start value is used with a counter overflow in order to calculate the new counter value.

### Parameter “Debounce time / min. operation 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 “Transmit counter values after bus voltage recovery”

If the parameter has the value “yes”, after bus voltage recovery (once the sending delay has elapsed) the current value of the meter is sent on the bus. If the intermediate meter has been enabled, these values will also be sent on the bus.

After an extended bus voltage failure, the meter is reset to the start value. If the intermediate meter has been enabled, it will also be set to zero. If no data loss has occurred with a brief drop in voltage, the meter readings are retained.

### Parameter “Enable additional options”

If this parameter is set to “yes”, the parameter window “A-Counter” becomes visible. Additional functions are possible here.

### Parameter window “A-Counter”

In this parameter window, additional functions for the counter can be enabled.

### Parameter “Divider: number of input pulses for one counter step”

Using this parameter you can define how many pulses are necessary to generate a metering pulse. It acts as a divider.

### Parameter “Factor: One counter step changes counter value by”

Using this parameter, you can determine by how much the meter reading and intermediate reading is incremented with a counter pulse. It acts as a multiplication factor.

### Parameter “Transmit counter values cyclically”

If this parameter has the value “yes”, the values of the counter and the intermediate counter are transmitted cyclically to the bus.

### Parameter “Counter values are being transmitted every”

This parameter is visible if the parameter “*Transmit counter values cyclically*” is equal to “yes”. You can set the intervals at which the values are transmitted cyclically on the bus.

## Commissioning

### Parameter "Enable differential counter"

Using this parameter, the object "Intermediate meter" is made visible. The intermediate meter can, for example, assume the function of a daily count meter.

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

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

In this parameter, you can set the value at which the intermediate (differential) counter generates an overflow. The same rules apply at overflow as apply with a normal meter. At overflow, the object "Differential counter overflow" is sent.

## 3.12.2 Communication objects

### Object "Meter reading ... byte": 1 to 4 byte

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

The following table provides an overview of the data types:

Data width	EIS type	Value range
1 byte	EIS 14	0...255
2 byte	EIS 10	-32.768...32.767
2 byte	EIS 10	0...65.535
4 byte	EIS 11	-2.147.483.648...2147483647

### Object "Differential counter ... bytes" 1 to 4 bytes

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

The object contains the state of the intermediate (differential) meter, which is identical in its counting function with the absolute counter. In contrast to this it can be reset (object "Reset differential counter") and a meter overflow can be reported on the bus (object "Differential counter overflow"). Using the differential counter, for example, daily consumption levels can be measured.

As soon as the differential meter reaches the overflow value determined in the parameter *"Over-/underrun of differential counter at"*, the overflow value is subtracted from the value of the differential meter.

### Object "Request counter values": 1 bit (EIS1)

Using this object, the values of the absolute meter and differential meter are requested.

Telegram value:      "0" no reaction  
                              "1" Send meter reading

### Object "Differential counter overflow": 1 bit (EIS1)

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

As soon as the value of the differential meter overruns or underruns the overflow value determined 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 (EIS1)

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

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

## 3.13 Programming

The device can be programmed with the software ETS. 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 once-off full download is required. This may take a few minutes.

**Note:** If an application program that is already programmed be reprogrammed, the device should be discharged using the ETS beforehand. This may be necessary in rare cases, e.g. if a fault has occurred during a download.

**Important:** If a device is not functional after programming, re-import the Universal Interface into the ETS and repeat the programming.

## Special functions

### 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.

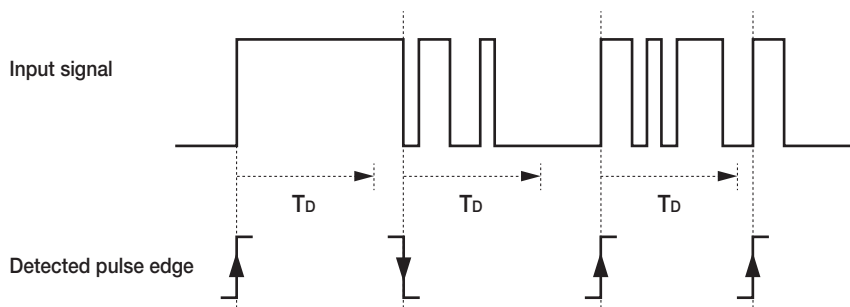
#### 4.1 Debounce time and minimum operation time

A debounce time or a minimum operation 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 the pulse edges are detected at the input during the debounce time they are ignored.

The following example makes this clear:



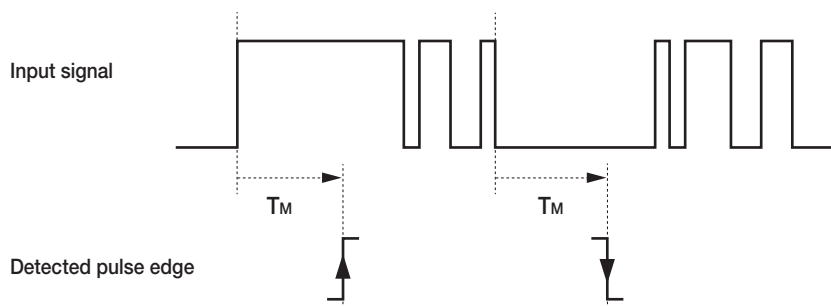
After detection of an edge on the input, further edges are ignored for the debounce time  $T_D$ .

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

#### 4.2 Limit rate of telegrams

A new monitoring period starts at the end of the previous monitoring period, or – in the case of a bus voltage recovery – at the end of the transmission delay time. The transmitted telegrams are counted. As soon as the “Max. number of sent telegrams is reached”, no further telegrams are sent on the bus until the end of the monitoring period. The telegram counter is reset to zero and sending of telegrams is allowed again with the start of a new monitoring period.

#### 4.3 Cyclic sending

Cyclic transmission is a part of the “Function Switch sensor”. It enables the 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 object in the setting have to be deleted to ensure that it cannot be 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 (after the transmission delay time has elapsed), the object value is sent immediately on the bus and the sending cycle time restarts. The minimum value for the transmission cycle time is 200 ms. If a smaller value is set in the parameters, the transmission cycle time is the same as the minimum value.

## Special functions

### 4.4 Dimming

As standard the co-called "1 button dimming" is preset, i.e. switching and dimming functions can be fully controlled via a single push button. For each dim operation, a BRIGHTER or DARKER dim telegram is sent alternately. If the communication object "Switch" = 0, a BRIGHTER telegram is sent. 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 of the dimming operation (sends dimming telegram)
OFF	DARKER	BRIGHTER
OFF	BRIGHTER	BRIGHTER
ON	DARKER	BRIGHTER
ON	BRIGHTER	DARKER

Table: Dimming function "1 button dimming"

If "2 button dimming" is required, the functions of the individual buttons should be set with the parameters "Reaction on short operation or Reaction on long operation", e.g. ON or BRIGHTER. The user has the freedom to choose,

- which push buttons are combined with one another to dim a lighting group and
- which function the individual push button uses in this case

### 4.5 Control scene

Using a scene, an input controls several actuator groups to a determined, preset value with just a single operation. Using the function "Control scene", the input can recall and/or store a scene. A scene can be implemented using two methods here.

#### Scene via 5 objects

On the one hand, each input can simultaneously store several actuator groups via 5 communication objects. When a scene is recalled, several telegrams are generally sent. Storing of a scene occurs by reading the current values of the actuator groups via the bus. An actuator group can either be controlled via 1 bit values (ON/OFF) or via 8 bit values (0...255 corresponding to 0...100%).

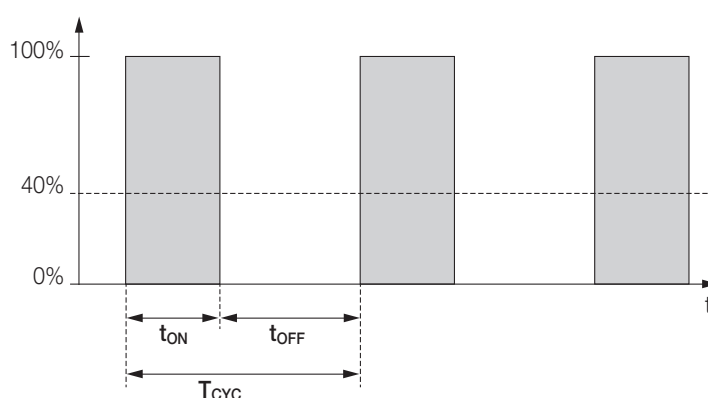
#### 8 bit scene

On the other hand, the input can be recalled by sending the object "Scene" values, which is stored in the actuator. This simplified scene function is only possible when it is supported by the actuator. An object "Scene" contains the scene number (1...64) and the information as to whether the scene should be recalled or stored. In the actuator, the scene number(s) to which it reacts is set.

### 4.6 Control electr. relay ("heating actuator")

The function "Heating actuator" switches an electronic relay, which is generally used to control an electro-thermal valve drive. The device is normally controlled by a room thermostat. Various types of control are possible (e.g. continuous-action control).

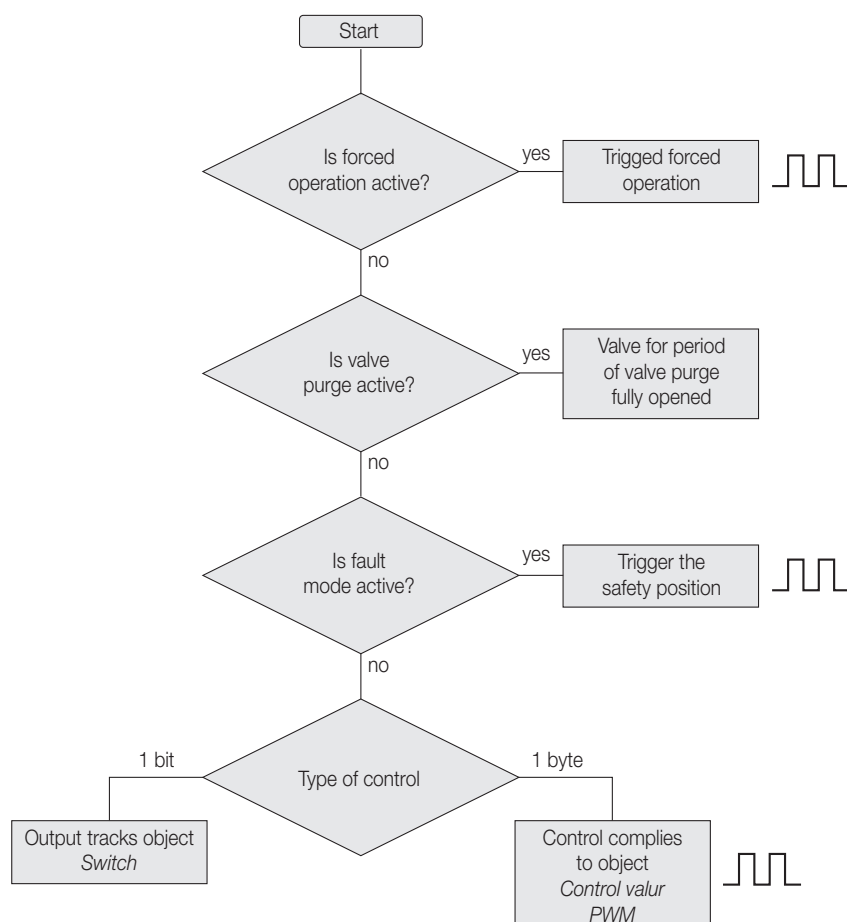
The electrothermal valve drive can be triggered via two-step control or pulse width modulation. With pulse width modulation, the control is implemented by a variable mark-space ratio. The following example clarifies this:



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

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

## Special functions



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:

Control of the valve via	Reaction on start	Reaction on end
Forced operation	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

The sequence in the table simultaneously defines the priority of the special positions. Forced operation has the highest priority.



## Special functions

### 4.7 Switching sequence

The function "Switching sequence" facilitates flexible switch on and off of up to five communication objects (1 bit) using just a single input.

A switching sequence consists of a sequence of switching steps that represent determined object values. An operation on the input increments or decrements a switching stage.

Example: Switching sequence "sequentially on/off (one push-button)" 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>...

The defined switching sequences stand out in that only the value of a single communication object changes between two switching stages. The following switching sequences are possible:

#### "Sequentially on/off (one push-button)"

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

#### "Sequentially on/off, multiple push-buttons"

The switching sequence is similar to the function "sequentially on/off (one push-button)" with the exception 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.

#### "All combinations (Gray code)"

In this sequence, all communication object combinations are undertaken successively. Only the value of a communication object is modified between two stages. 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 with the Gray code can be found in the appendix under section 6.1.

#### Other combinations

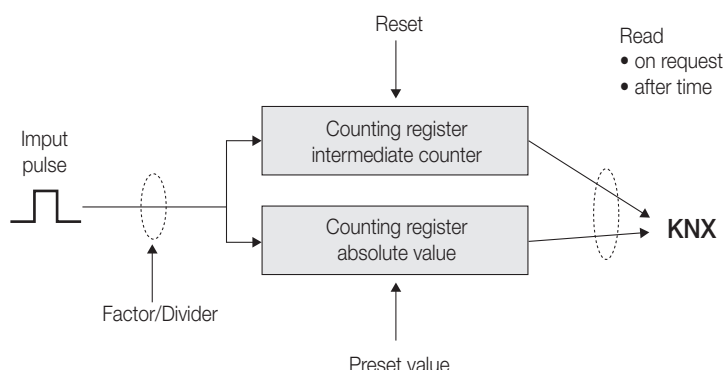
As well as operation via the input, the switching stage can be modified via communication objects "Level increment/decrement". This is used, for example, to increment or decrement two or three inputs. When a value is received on this communication object, the response is identical to the operation of the input.

**Note:** The current switching step always results from the state of the communication objects. If, for example, a communication object of another device changes, the current switching step can also change in this way.

## Special functions

### 4.8 Count pulses

The meter function is used for counting binary switching pulses. It contains all the important features necessary for high-performance meters. The following diagram provides an overview:



In addition to the absolute meter, a differential meter can be enabled that measures differential values (e.g. comparable with a trip recorder). In order to adapt the counting speed, the factors and dividers can be set commonly for both meters.

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

The overflow of the differential meter can be sent via telegram on the bus. This telegram can be used to read out the absolute meter readings. Thus, the absolute meter reading is always sent after an adjustable change on the bus.

A bus voltage failure can lead to deleting of the counter values (meter readings).

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

**Note:** The connection of the US/U x.2 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).

### 4.9 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.

### 4.10 Reaction after bus voltage recovery

In this section, the response of the device after bus voltage recovery with initialization is described.

After bus voltage recovery, the device always waits for the send delay time to elapse before sending telegrams on the bus. The transmission delay time can be set in the general parameters.

The following diagram indicates the time sequence:

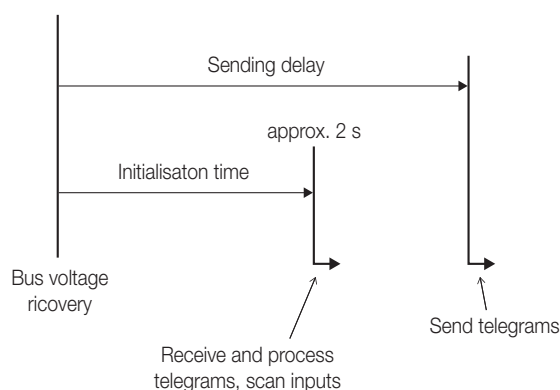


Figure 2: Reaction after bus voltage recovery

## Special functions

On bus voltage recovery, the inputs are scanned after the initialisation period and the object values are updated when possible. If the input is operated, the device will behave as if the operation commenced at the end of the initialisation time.

The reaction depends on the function of the channel. The following list provides an overview:

Function	Reaction after bus voltage recovery*
Switch sensor	If a distinction is made between short and long operation, or if the "TOGGLE" value is set in one of the "Reaction on closing/opening the contact" parameters, no telegram will be sent on bus voltage recovery. Otherwise, the reaction can be set in the parameters..
Switch/dimming sensor	No telegram is sent on the bus.
Shutter sensor	No telegram is sent on the bus.
Value / forced operation	Object values are overwritten (updated) by the parameterized values.
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 parameterized value is set.
Control LED	The output state of the output can be set in the parameters.
Switching sequence ("impulse relay")	No telegram is sent on the bus.
Push-button with multiple operation	No telegram is sent on the bus.
Counter	No telegram is sent on the bus.

\*More exact response directly after the transmission delay

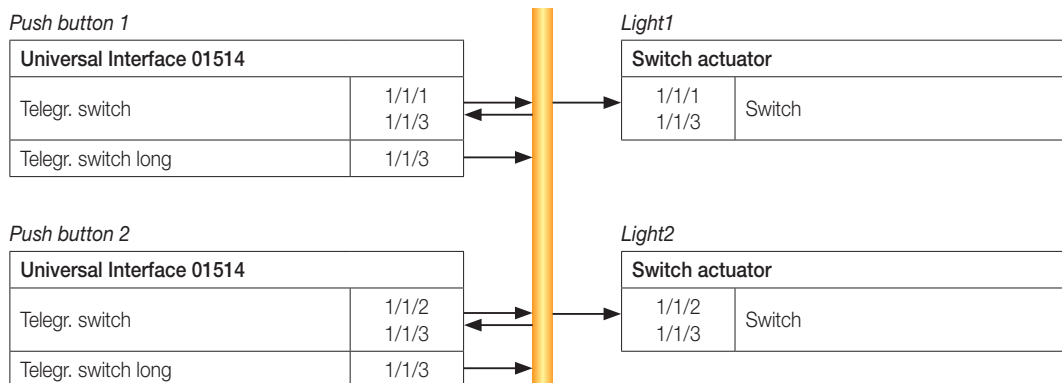
## Planning and application

### 5. Planning and application

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 (Switch light)

The lighting is switched on and/or off with a short push of the button. A long button push switches the light off centrally.  
Logical connection of the group addresses:



Parameter settings for push button 1 and 2:

General	
Channel A	Function of the channel
Channel B	Distinction between long and short operation
Channel C	Connected contact type
Channel D	Reaction on short operation
	Reaction on long operation
	Long operation after: Base
	Factor [2...255]
	Number of objects for short/long operation
	Debounce time

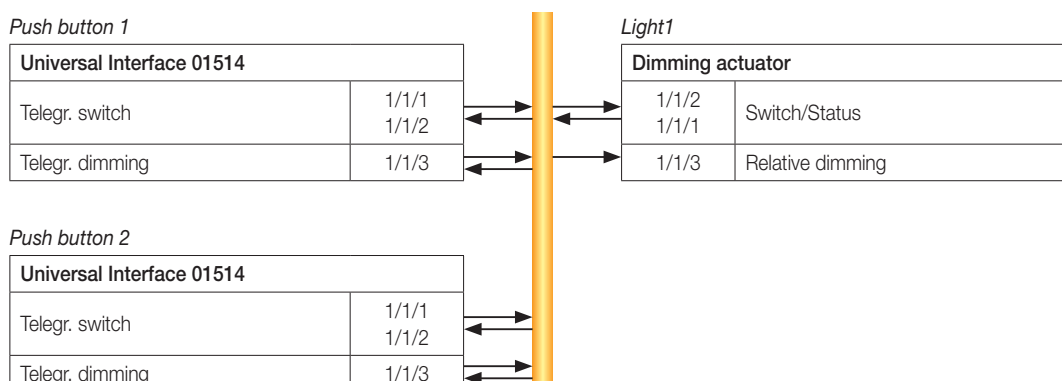
Switch sensor
yes
normally closed
ON
OFF
100ms
5
1 object
50ms debounce time

#### 5.2 Operation of dimmable illumination

##### 1 button operation

Short operation switches on or off the lighting, a longer operation alternately dims BRIGHTER or DARKER (contrary to the last dimming process). Both buttons operate the same lighting.

Logical connection of the group addresses:



## Planning and application

Parameter settings for push button 1 and 2:

General	Function of the channel	Switch/dimming sensor
Channel A	Connected contact type	normally closed
Channel B	Dimming functionality	Dimming and switching
Channel C	Reaction on short operation	TOGGLE
Channel D	Reaction on long operation	Dim BRIGHTER/DARKER
	Dimming direction after switching ON	DARKER
	Long operation after	0.5s
	Dimming mode	Start-stop-dimming
	Debounce time	50ms debounce time

### 2 button operation

The same group address logical connection is also suitable for 2 button dimming. Modification of the parameters:

"Reaction on short operation" = "ON" or "OFF"

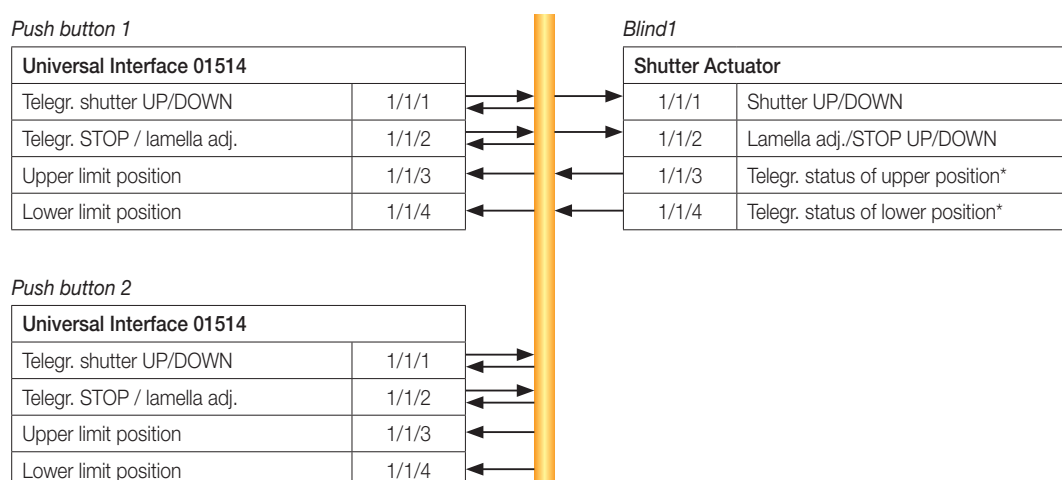
"Reaction on long operation" = "Dim BRIGHTER" or "Dim DARKER".

## 5.3 Operation of blinds

### 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:



## Planning and application

Parameter settings for push button 1 and 2:

General	
Channel A	Function of the channel
Channel B	Operating functionality of blind
Channel C	Long operation: Lamella Short operation: Move UP/DOWN
Channel D	Connected contact type
	Long operation after
	"Telegr. STOP/lamella adj." is repeated every
	Debounce time

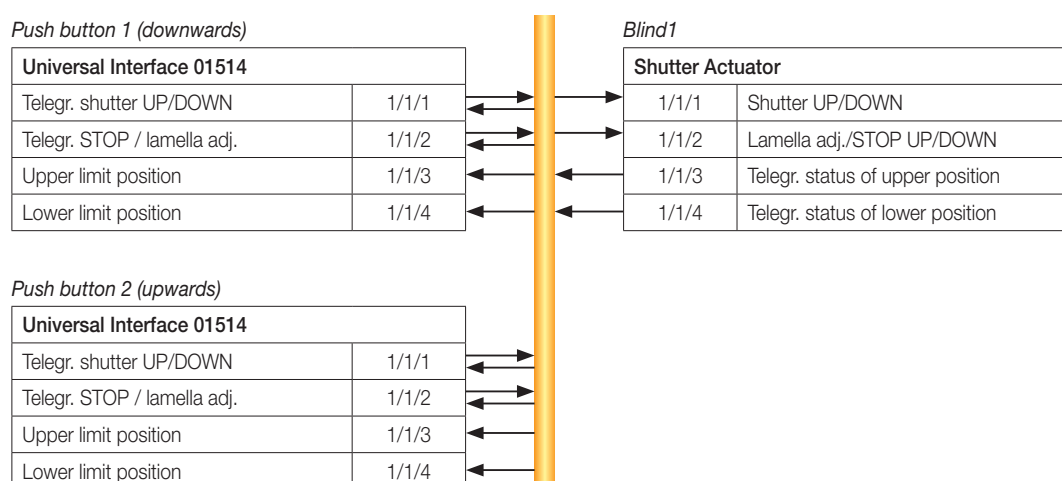
Shutter sensor  
1-push-button, short = moving, long = stepping  
<-- Note about functionality  
normally closed  
0.5s  
0.4s  
30ms

\* 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:

General	
Channel A	Function of the channel
Channel B	Operating functionality of blind
Channel C	Short operation: STOP / lamella UP/DOWN Long operation: move UP/DOWN
Channel D	Connected contact type
	Reaction on short operation
	Reaction on short operation
	Long operation after
	Debounce time

Shutter sensor  
2-push-button, standard  
<-- Note about functionality  
normally closed  
STOP / lamella DOWN  
MOVE DOWN  
0.5s  
30ms

## Planning and application

Parameter settings for push button 2:

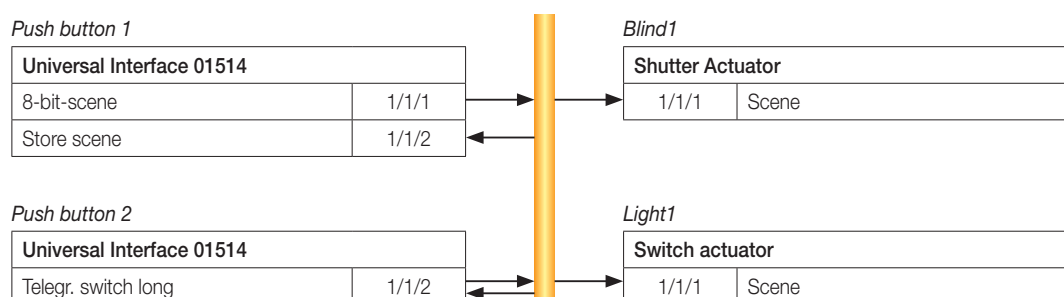
Reaction on short operation	STOP / lamella UP
Reaction on long operation	MOVE UP

### 5.4 Control of scenes

#### 8 bit scene\*

Push button 1 and push button 2 control Shutter 1 and Light1. Push button 1 recalls the scene. On a long operation of push button 2 the current shutter position and the state of the lighting are stored. Storage is undertaken in the actuator.

Logical connection of the group addresses:



Parameter settings for push button 1:

<b>General</b>		
<b>Channel A</b>	Function of the channel	Control scene
Channel B	Connected contact type	normally closed
Channel C	Control the scene via	8-bit-scene
Channel D	No. of scene	Scene no. 9
	Reaction on short operation	Recall scene
	Store scene	with object value = 1
	Debounce time	50ms debounce time

Parameter settings for push button 2:

<b>General</b>		
<b>Channel A</b>	Function of the channel	Switch sensor
Channel B	Distinction between long and short operation	yes
Channel C	Connected contact type	normally closed
Channel D	Reaction on short operation	no reaction
	Reaction on long operation	OFF
	Long operation after:	100ms
	Base	
	Factor [2...255]	30

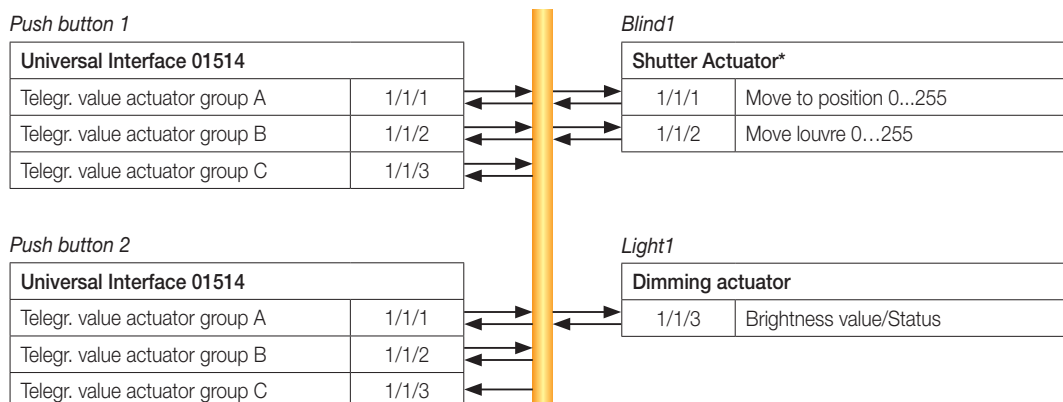
\* The 8 bit scene requires actuators that support these functions. This is the case for ABB shutter actuators and switch actuators of the latest generation (from 2003). On other devices the scene is recommended via "5 separate objects".

## Planning and application

### Scene via 5 separate objects

Push button 1 and push button 2 control Shutter 1 and Light1. Short operation recalls the scene. On long operation, the current shutter position and the brightness value are stored. Both push buttons store different scene values.

Logical connection of the group addresses:



Parameter settings for push button 1 and 2:

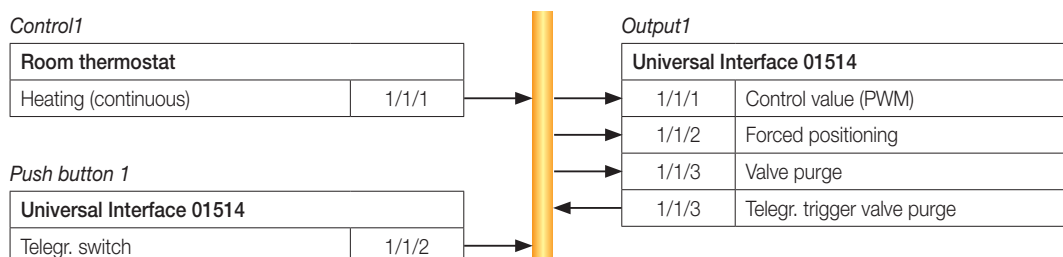
General	
<b>Channel A</b>	Function of the channel
A-Scene	Control scene
Channel B	Connected contact type
Channel C	normally closed
Channel D	Control the scene via
	5 separate objects
	Reaction on short operation
	Recall scene
	Store scene
	on long operation
	Long operation after
	3s
	Debounce time
	50ms debounce time

\* 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

On Output1, a Universal Interface is connected to an Electronic Relay ER/U 1.1, which controls an electro-thermal valve drive. The room temperature is continuously controlled via Control1. The valve is purged once a week by opening it for approx. 5 minutes. The valve can be forcibly fully opened via push button 1. If a telegram is not received for 30 minutes from Control1, the valve will be opened to 30 % (fault mode).

Logical connection of the group addresses:





## Planning and application

Parameter settings for output 1:

General	Function of the channel	Control electr. relay (heating actuator)
Channel A	Control telegram is received as	1 byte (continuous)
A-Fault/Forced Operat.	Connected valve type	normally closed
Channel B	PWM-cycle time for continuous control	1min
Channel C	Enable object "Telegr. valve purge"	yes
Channel D	Enable monitoring of the controller, fault message, forced positioning	yes
	Position of the valve drive on bus voltage recovery	30%

General	Monitoring of the room thermostat	yes
Channel A	Cyclic monitoring time of room thermostat: base	1min
A-Fault/Forced Operat.	Factor [1...255]	30
Channel B	Position of the valve drive on failure of the control	30%
Channel C	Enable object "Telegr. fault"	no
Channel D	Forced positioning	yes
	valve position during forced postioning	100% (opened)

General	Transmission delay [2...255s] after bus voltage recovery	2
Channel A	The transmission delay time contains the initialization time	<--- NOTE
A-Fault/Forced Operat.	Limit number of telegrams	no
Channel B	Transmit object "Telegr. valve purge"	yes
Channel C	Transmit telegram every	7 days
Channel D	Period of valve purge	5min

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

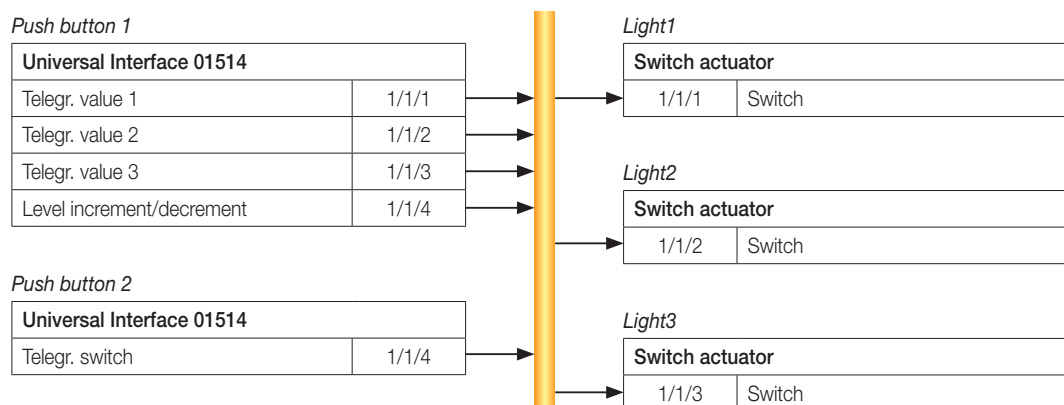
## Planning and application

### 5.6 Switching of lighting in switching sequence

#### Successive switch on/off

Push button 1 and push button 2 control a lamp with two independent circuits Light1, Light2 and Light3. Push button 1 switches on successively on operation (sequence: Light1>Light2>Light3). Push button 2 switches off successively on operation (sequence: Light3>Light2>Light1).

Logical connection of the group addresses:



Parameter settings for push button 1:

<b>General</b>	
Channel A	Function of the channel
Channel B	Connected contact type
Channel C	Number of objects
Channel D	Type of switching sequence
	Example for switching sequence
	Function on operation
	Debounce time / min. operation time

Switching sequence ("latching relay")  
normally closed  
3 levels  
sequentially on/off (one push-button)  
<--- NOTE  
switch upwards  
50ms debounce time

Push button 2 has to be parameterized so that "Telegr. switch" sends a "0" with every button push.

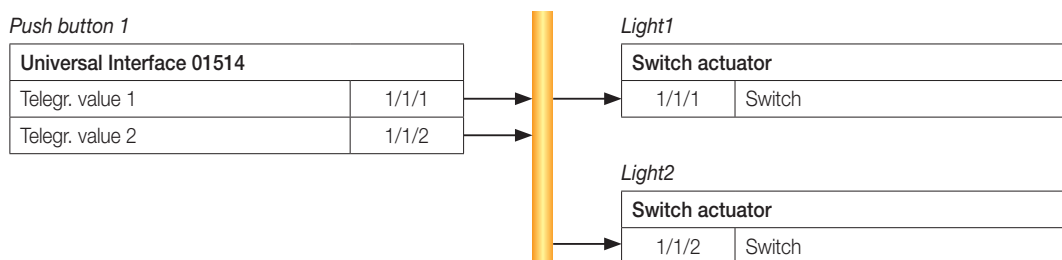
#### Switch all combinations (Gray code)

Push button 1 controls a lamp with two independent circuits Light 1 and Light 2. On operation, 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)		

## Planning and application

Logical connection of the group addresses:



Parameter settings for push button 1:

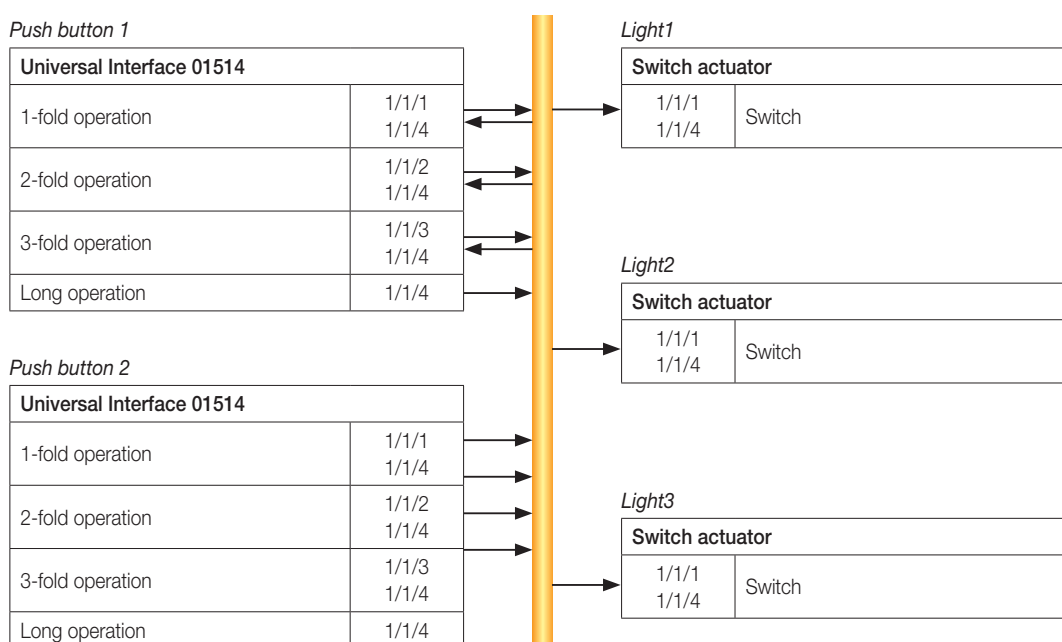
General	
Channel A	Function of the channel
Channel B	Connected contact type
Channel C	Number of objects
Channel D	Type of switching sequence
	Example for switching sequence
	Debounce time / min. operation time

Switching sequence ("latching relay")
normally closed
2 levels
All combinations
<--- NOTE
50ms debounce time

### 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:



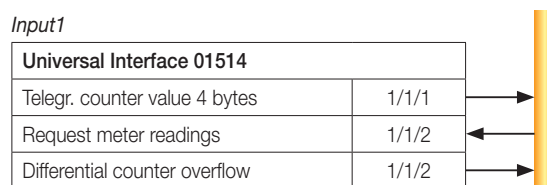
## Planning and application

Parameter settings for push button 1 and 2:

General		
Channel A	Function of the channel	Push-button with multiple operation
Channel B	Connected contact type	normally open
Channel C	Max. number of operations ( = number of objects)	3-fold operation
Channel D	Transmitted value (object "Telegr. operation ..-fold")	TOGGLE
	Transmit value on every operation	no
	Maximum time between two operations	1s
	Additional object for long operation	yes
	Long operation after	0.5s
	transmitted value (object "Telegr. operation long")	TOGGLE
	Debounce time	50ms

### 5.8 Metering energy values

Input1 is connected to the S0 output of an energy meter (100 Impulse/kWh). The 4 byte meter value is represented on the bus in the Wh unit. It is sent on the bus every 30 seconds as well as with every change of 100 Wh.



Parameter settings for input1:

General		
Channel A	Function of the channel	Counter
A-Counter	Pulse detection on	closing contact (rising edge)
Channel B	Data width of counter	32-bit [-2.147.483.648 ... 2.147.483.647]
Channel C	Counter starts at [-2.147.483.648...2.147.483.647]	0
Channel D	Debounce time / min. operation time	50ms debounce time
	The debounce time must be shorter than the pulse period of the input signal	<--- NOTE
	Transmit counter values after bus voltage recovery	no
	Enable additional options (factor/divider, cyclical transmission)	yes

## Planning and application

General	Divider: number of input pulses for one counter step [1...32767]	1
Channel A	Factor: One counter step changes counter value by [-32768...32767]	10
<b>A-Counter</b>	Transmit counter values cyclically	yes
Channel B	Counter values are being transmitted every: Base	1s
Channel C	Factor [1...255]	30
Channel D	Enable differential counter	yes
	Over-/underrun of differential counter at [-2147483648...2147483647]	1000
	The overrun value must be greater than the factor	<--- NOTE

100 pulses per kWh means 1 pulse per 10 Wh. The meter is therefore incremented by the value 10 for every pulse (factor = 10).

The differential counter overflows every 100 Wh. At every overflow, the object "Differential counter overflow" object is incremented by the value "1". This is received by the object "Request meter readings" and the current meter reading is sent on the bus.

## Appendix

### 6. Appendix

#### 6.1 Gray code table

The sequence is characterized by the fact that only one value changes between two steps. Thus, the transition to the next step only requires sending a single telegram.

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

Switching stage		Value of the communication objects				
No.	Short designation	"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	ON	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





01514 EN 01 1701



**VIMAR**

Viale Vicenza 14  
36063 Marostica VI - Italy  
[www.vimar.com](http://www.vimar.com)