ABB i-bus® KNX
Room Master RM/S 4.1
Product Manual

Power and productivity
for a better world™

Contents

Page

1 General

The Room Master RM/S 4.1 provides intelligent engineering technology for different room layouts and configurations, e.g. for hotel rooms and apartments.

Modern buildings require intelligent building engineering technology for safe and efficient operation. Many buildings world-wide already utilise the full potential of networked electrical installations.

Hotels, hospitals, senior citizen and student residential homes, assisted living accommodation and much, much more: The Room Master covers new possibilities for buildings in the residential, purpose-built and hotel sectors.

The Room Master has been developed for all rooms of this type. It covers all requirements of the electrical installation of this application and offers the following functions in compact form:

- Switching lighting
- Switching of loads

In addition to these basic functions, further automation functions can be implemented by a combination with a presence detector. The communication of the devices via the KNX bus also enables control functions as well as sending of emergency signals from the rooms to a control centre.

1.1 Using the product manual

This manual provides you with detailed technical information relating to the function, installation and programming of the ABB i-bus® KNX Room Master RM/S 4.1. The application of the device is explained using examples.

This manual is subdivided into the following chapters:

Chapter 1	General
Chapter 2	Device technology
Chapter 3	Commissioning
Chapter 4	Planning and application
Chapter 5	Room scenarios
Chapter A	Appendix

1.1.1 Notes

Notes and safety instructions are represented as follows in this manual:

Note
Tips for usage and operation

Examples
Application examples, installation examples, programming examples

Important
These safety instructions are used as soon as there is danger of a malfunction without risk of damage or injury.

Caution
These safety instructions are used as soon as there is danger of a malfunction without risk of damage or injury.

⚠ **Danger**
These safety instructions are used if there is a danger for life and limb with inappropriate use.

⚠ ⚡ **Danger**
These safety instructions are used if there is a danger to life with inappropriate use.

1.2 Room Master: Areas of application

1.2.1 Residential care homes

The Room Master facilitates comfort and security in residential care homes and supports senior citizens in their daily routine:

- Simple operation of the room functions
- Automatic transmission of messages to the control station, e.g. emergency signals
- Fast localisation of faults in the rooms
- Indication of room states in the control station
- Day/night service

1.2.2 Apartments

Apartments gain in both their appeal and the standard of living they offer with the Room Master – decisive factors for sale and rental:

- Automatic switching of different lighting arrangements in the room
- Comfortable and simple operation of the room functions

1.2.3 Hospitals

When used in hospitals and buildings with a similar purpose, the Room Master features many functions which support the efficient running of a modern operation:

- Simple operation of the room functions
- Day/night service
- Indication of the ward round
- Remote control of the room and display of the room state in the nurses station
- Fast localization of faults in the rooms and simplified room maintenance

1.2.4 Hotel

The Room Master offers all functions which are required in a modern hotel room. During operation, a range of advantages are achieved in comparison to a conventional installation:

- Comfortable and simple operation
- Transmission of messages
- Fast localisation of faults

The advantages of the Room Master are obvious not just during operation, but also for planning:

- World-wide use
- Compact design
- A standard solution for many projects.

1.3 Product and functional overview

The Room Master RM/S is used as a single room solution. The RM/S is used to control the lighting. The input signals are detected via binary inputs or directly via the sensors connected to the KNX.

Management systems can directly access the RM/S via the ABB i-bus® and activate controls in the room.

The Room Master is a modular installation device with a module width of 8 space units in Pro *M* design for installation in the distribution board. The connection to the ABB i-bus® is established using the front side bus connection terminal. The device can be operated manually, for example, during commissioning, by applying an auxiliary voltage to the bus terminals. The assignment of the physical addresses as well as the parameterization is carried out with Engineering Tool Software ETS.

The device features eight switching outputs for control of lighting, e.g. such as

- Lighting in the room
- Bathroom and entrance lighting

Furthermore, eight floating binary inputs are available. These are used to report room information to the Room Master, e.g. switch light ON/OFF:

- Room lighting
- Bathrom lighting
- Transmission of an emergency signal

Higher-level room scenarios can also be programmed.

The scanning voltage for the binary inputs is provided by the device. The binary inputs are divided into four groups of two inputs each.

Overview of the number and allocation of the inputs and outputs:

Inputs	RM/S 4.1
Binary via contact scanning	8

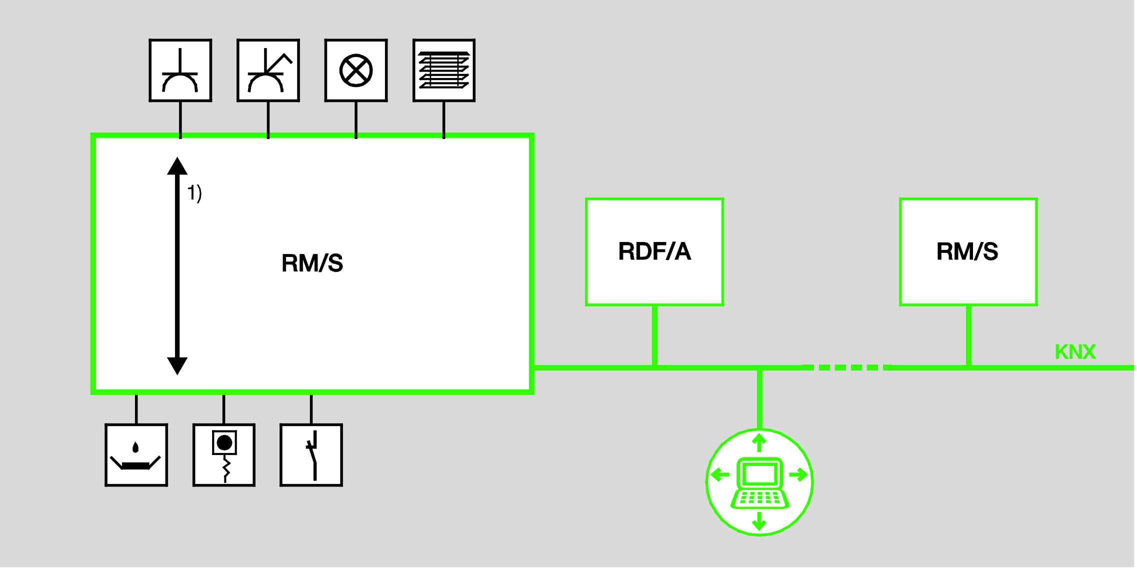
Outputs	RM/S 4.1
Switching contact 6 A	8

1.4 Function of the Room Scenarios

With the innovative concept of the Room Master RM/S, it is possible to recall the entire Room Scenarios with just one group address. The recall of a Room Scenario can be undertaken both internally, e.g. via a binary input as well as externally. The recalled room state sets the outputs via KNX scenes. These can also be internally or externally called.

After the room state is recalled, all functions in the room, e.g. illumination, energy supply, are adapted to the parameterization settings.

The Room Master features internal device interconnections between the inputs and outputs. No group addresses are required for internal communication. This prevents an unnecessary bus load.



1 Internal device connections

2 Device technology

2CDC 071 020 S0012

RM/S 4.1

The Room Master is a modular installation device (MDRC) in Pro *M* design. It is intended for installation in the distribution board on 35 mm mounting rails. The assignment of the physical addresses as well as the parameterization is carried out with the ETS and the current application.

The RM/S is powered via the ABB i-bus® and does not require and additional auxiliary voltage supply.

The device is ready for operation after connecting the bus voltage.

2.1 Technical data

Supply	Bus voltage	21…32 V DC
	Current consumption, bus	Maximum 12 mA (Fan-in 1)
	Leakage loss, bus	Maximum 250 mW
	Leakage loss, device	Maximum 1.68 W *
* The maximum power consumption of the device results from the following specifications:	Relay 6 A	1.6 W
Connections	KNX	Via bus connection terminals, 2-fold (red/black) 0.8 mm Ø, solid
	Circuits	Screw terminal with universal head (PZ 1) 0.2…4 mm² stranded, 2 x (0.2…2.5 mm²) 0.2…6 mm² single core, 2 x (0.2…4 mm²)
	Ferrules without/with plastic sleeves	without: 0.25…2.5 mm² with: 0.25…4 mm²
	TWIN ferrules	0.5…2.5 mm²
	Tightening torque	Maximum 0.6 Nm
Operating and display elements	Button/LED ○ ●	For assignment of the physical address
Enclosure	IP 20	Compliant to EN 60 529
Safety class	II	Compliant to EN 61 140
Insulation category	Overvoltage category	III to EN 60 664-1
	Pollution degree	2 to EN 60 664-1
KNX safety extra low voltage	SELV 24 V DC	

Temperature range	Operation Transport Storage	-5 °C…+45 °C -25 °C…+70 °C -25 °C…+55 °C
Ambient conditions	Maximum air humidity	93 %, no condensation allowed
Design	Modular installation device (MDRC)	Modular installation device, Pro *M*
	Dimensions	90 x 144 x 64.5 mm (H x W x D)
	Mounting width in space units	8 modules at 18 mm
	Mounting depth	64.5 mm
Installation	On 35 mm mounting rail	Compliant to EN 60 715
Mounting position	as required	
Weight	0.3 kg	
Housing/colour	Plastic housing, grey	
Approvals	KNX to EN 50 090-1, -2	Certification
CE mark	In accordance with the EMC guideline and low voltage guideline	

Important

The maximum permissible current of a KNX line may not be exceeded.
During planning and installation ensure that the KNX line is correctly dimensioned.
The device features a maximum current consumption of 12 mA (Fan-In 1).

2.1.1 Binary inputs

Rated values	Number	8[1)]
	U_n scanning voltage	32 V, pulsed
	I_n scanning current	0.1 mA
	Scanning current I_n at switch on	Maximum 355 mA
	Permissible cable length	≤ 100 m one-way, at cross-section 1.5 mm² even when the core is routed in a multi-control cable

[1)] All binary inputs are internally connected to the same potential.

2.1.2 Rated current output 6 A

Rated values	Number	8 contacts
	U_n rated voltage	250/440 V AC (50/60 Hz)
	I_n rated current (per output)	6 A
Switching currents	AC3* operation (cos φ = 0.45) To EN 60 947-4-1	6 A/230 V
	AC1* operation (cos φ = 0.8) To EN 60 947-4-1	6 A/230 V
	Fluorescent lighting load to EN 60 669-1	6 A/250 V (35 µF)[2]
	Minimum switching power	20 mA/5 V 10 mA/12 V 7 mA/24 V
	DC current switching capacity (resistive load)	6 A/24 V=
Service life	Mechanical service life	$> 10^7$
	Electronic service life To IEC 60 947-4-1	
	AC1* (240 V/cos φ = 0.8)	$> 10^5$
	AC3* (240 V/cos φ = 0.45)	$> 1.5 \times 10^4$
	AC5a* (240 V/cos φ = 0.45)	$> 1.5 \times 10^4$
Switching times[1]	Maximum relay position change per output and minute if only one relay is switched.	2,683

[1] The specifications apply only after the bus voltage has been applied to the device for at least 10 seconds. Typical delay of the relay is approx. 20 ms.

[2] The maximum inrush-current peak may not be exceeded, see chapter 2.1.3.

*** What do the terms AC1, AC3 and AC5a mean?**

In Intelligent Installation Systems, different switching capacity and performance specifications, which are dependent on the special application, have become established in industrial and residential systems. These performance specifications are rooted in the respective national and international standards. The tests are defined so that typical applications, e.g. motor loads (industrial) or fluorescent lamps (residential) are simulated.

The specifications AC1 and AC3 are switching performance specifications which have become established in the industrial field.

Typical application:

AC1 –	Non-inductive or slightly inductive loads, resistive furnaces (relates to switching of ohmic/resistive loads)
AC3 –	Squirrel-cage motors: Starting, switching off motors during running (relates to (inductive) motor load)
AC5a –	Switching of electric discharge lamps

These switching performances are defined in the standard EN 60947-4-1 *Contactors and motor-starters - Electromechanical contactors and motor-starters*. The standard describes starters and/or contactors that previously were preferably used in industrial applications.

2.1.3 Output lamp load 6 A

Lamps	Incandescent lamp load	1200 W
Fluorescent lamps T5/T8	Uncorrected Parallel compensated DUO circuit	800 W 300 W 350 W
Low-voltage halogen lamps	Inductive transformer Electronic transformer	800 W 1000 W
	Halogen lamps 230 V	1000 W
Dulux lamp	Uncorrected Parallel compensated	800 W 800 W
Mercury-vapour lamp	Uncorrected Parallel compensated	1000 W 800 W
Switching capacity (switching contact)	Maximum peak inrush-current Ip (150 s)	200 A
	Maximum peak inrush-current I_p (250 µs)	160 A
	Maximum peak inrush-current I_p (600 µs)	100 A
Number of electronic ballasts (T5/T8, single element)[1)]	18 W (ABB EVG 1 x 18 CF)	10
	24 W (ABB EVG-T5 1 x 24 CY)	10
	36 W (ABB EVG 1 x 36 CF)	7
	58 W (ABB EVG 1 x 58 CF)	5
	80 W (Helvar EL 1 x 80 SC)	3

[1)] For multiple element lamps or other types, the number of electronic ballasts must be determined using the peak inrush current of the electronic ballasts.

Device type	Application	Max. number of communication objects	Max. number of group addresses	Max. number of associations
RM/S 4.1	Room Master 4/…*	255	255	255

* … = current version number of the application. **Please observe the software information on our homepage for this purpose.**

Note

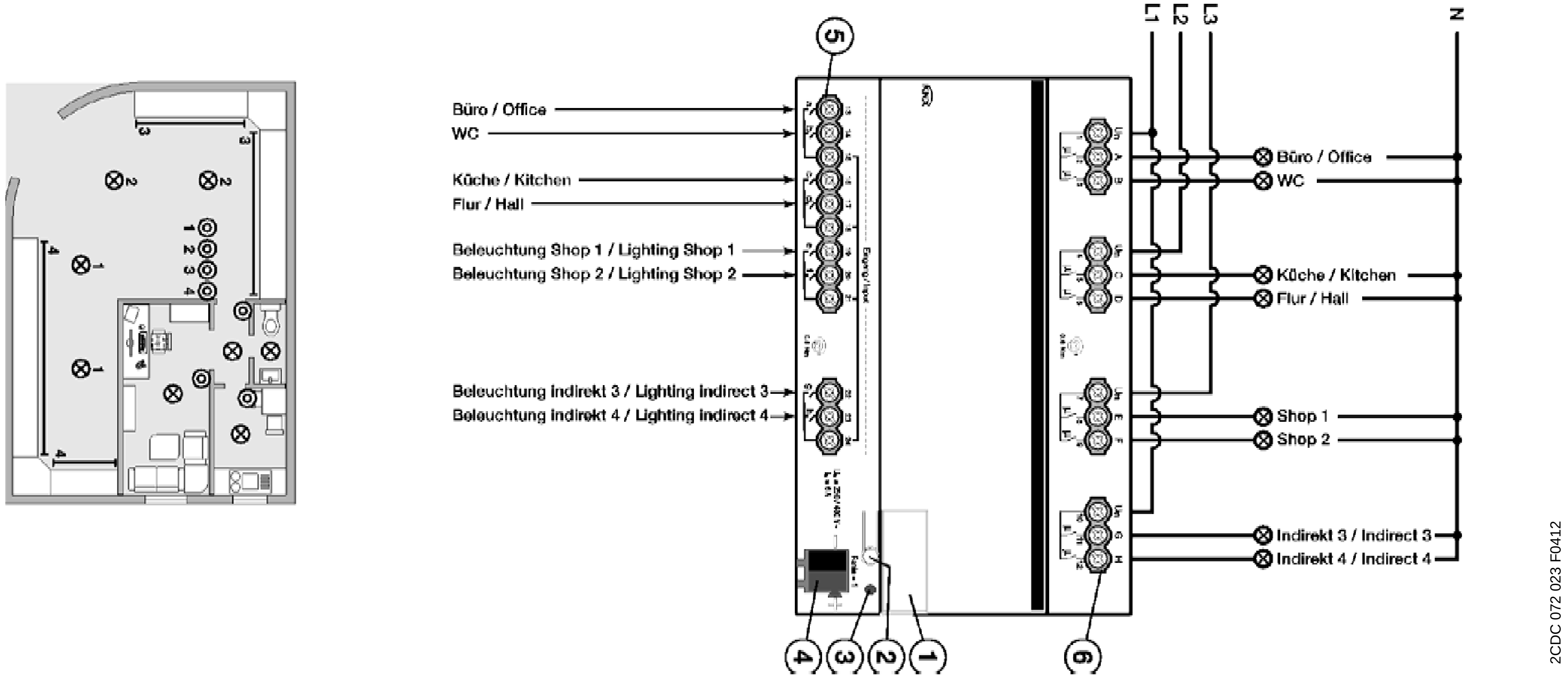
The ETS and the current version of the device application are required for programming.

The current version of the application is available for download on the internet at *www.abb.com/knx*. After import it is available in the ETS under *ABB/Room automation/Room Master*.

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

2.2 Connection schematic

Shop example



RM/S 4.1

1 Label carrier

2 Button *Programming*

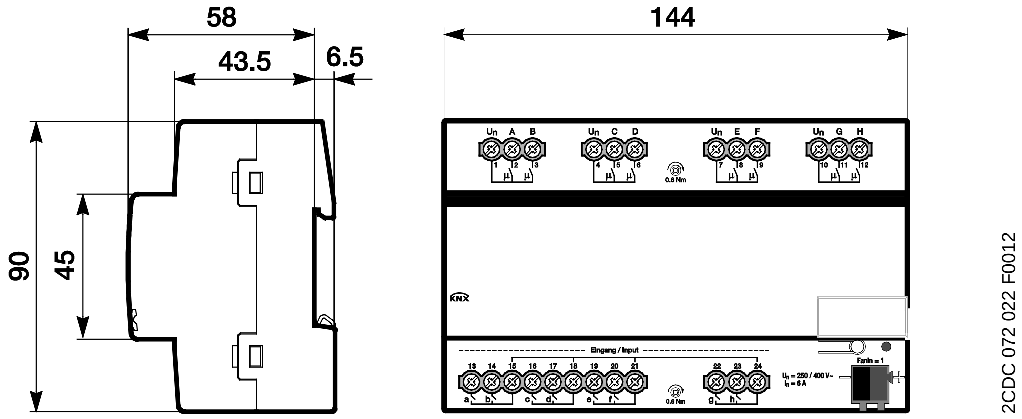
3 LED *Programming* ● (red)

4 Bus connection terminal

5 Binary inputs (a, b, c, d, e, f, g, h)

6 Load circuits, with 2 terminals each

2.3 Dimension drawing



2.4 Assembly and installation

The device is a modular installation device for quick installation in the distribution board on 35 mm mounting rails to EN 60 715.

The mounting position can be selected as required.

The electrical connection is implemented using screw terminals. The connection to the bus is implemented using the supplied bus connection terminal. The terminal assignment is located on the housing.

The device is ready for operation after connection to the bus voltage.

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

Commissioning requirements

In order to commission the device, a PC with ETS as well as an interface to the ABB i-bus®, e.g. via a KNX interface, is required.

The device is ready for operation after connection to the bus voltage supply. No additional auxiliary voltage is required.

Important
The maximum permissible current of a KNX line may not be exceeded. During planning and installation ensure that the KNX line is correctly dimensioned. The device features a maximum current consumption of 12 mA (Fan-In 1).

The installation and commissioning may only be carried out by electrical specialists. The appropriate norms, guidelines, regulations and specifications for your country should be observed when planning and setting up electrical installations and security systems for intrusion and fire detection.

- Protect the device from damp, dirt and damage during transport, storage and operation.
- Only operate the device within the specified technical data limits!
- The device should only be operated in an enclosed housing (distribution board)!
- The voltage supply to the device must be switched off, before mounting work is performed.

⚠ ⚠ **Danger**
In order to avoid dangerous touch voltages, which originate through feedback from differing phase conductors, all-pole disconnection must be observed when extending or modifying the electrical connections.

Supplied state

The device is supplied with the physical address 15.15.255. The application is pre-installed. It is therefore only necessary to load group addresses and parameters during commissioning.

However, the complete application can be reloaded if required. A longer downtime may result if the application is changed or after a discharge.

Assignment of the physical address

The assignment and programming of the physical address is carried out in the ETS.

The device features a button for assignment of the physical device address ⊸. The red LED ● lights up, after the button has been pushed. It switches off, as soon as the ETS has assigned the physical address or the button ⊸ has been pressed again.

Download response

Depending on the PC which is used, the progress bar for the download may take up to one and a half minutes, before it appears, due to the complexity of the device.

Cleaning

The voltage supply to the device must be switched off before cleaning. If devices become dirty, they can be cleaned using a dry cloth or a cloth dampened with a soapy solution. Corrosive agents or solutions should never be used.

Maintenance

The device is maintenance-free. No repairs should be carried out by unauthorised personnel if damage occurs, e.g. during transport and/or storage.

3 Commissioning

3.1 Overview

The parameterization of the Room Master is implemented with the application program *Room Master 4/1* and the Engineering Tool Software ETS. Using the application, a comprehensive and flexible range of functions are available to the device. The standard settings allow simple commissioning. The functions can be extended if required.

The following functions are available:

Lighting	For supply of eight lighting circuits, e.g. room, bathroom, hall, entrance area.
Binary input	8 binary inputs are available, e.g. Light ON/OFF switching in the entrance area of the room, in the bathroom, the freestanding or table lamps, and sending of an emergency signal.

The Room Master features relays in each switching output, which are mechanically independent of the other outputs. Switching noises cannot be avoided due to the mechanical nature of the design.

The device is installed primarily in the distribution board together with the circuit-breakers and RCCBs.

3.1.1 Functions of the inputs

The following table provides an overview of the functions possible with the inputs of the device and the application *Room Master*:

Functions of the inputs	a…h
Switch sensor	(
Dim sensor	(
Blind sensor	(
Value / forced operation	(

■ = Function is supported

3.1.2 Copying and exchanging parameter settings

Parameterization of devices can take a lot of time depending on the complexity of the application and the number of device inputs/outputs. To keep the commissioning work to the minimum possible, using the function *Copy/exchange channels*, parameter settings of an output can be copied or exchanged with freely selectable outputs. Optionally, the group addresses can be retained, copied or deleted in the target input/output.

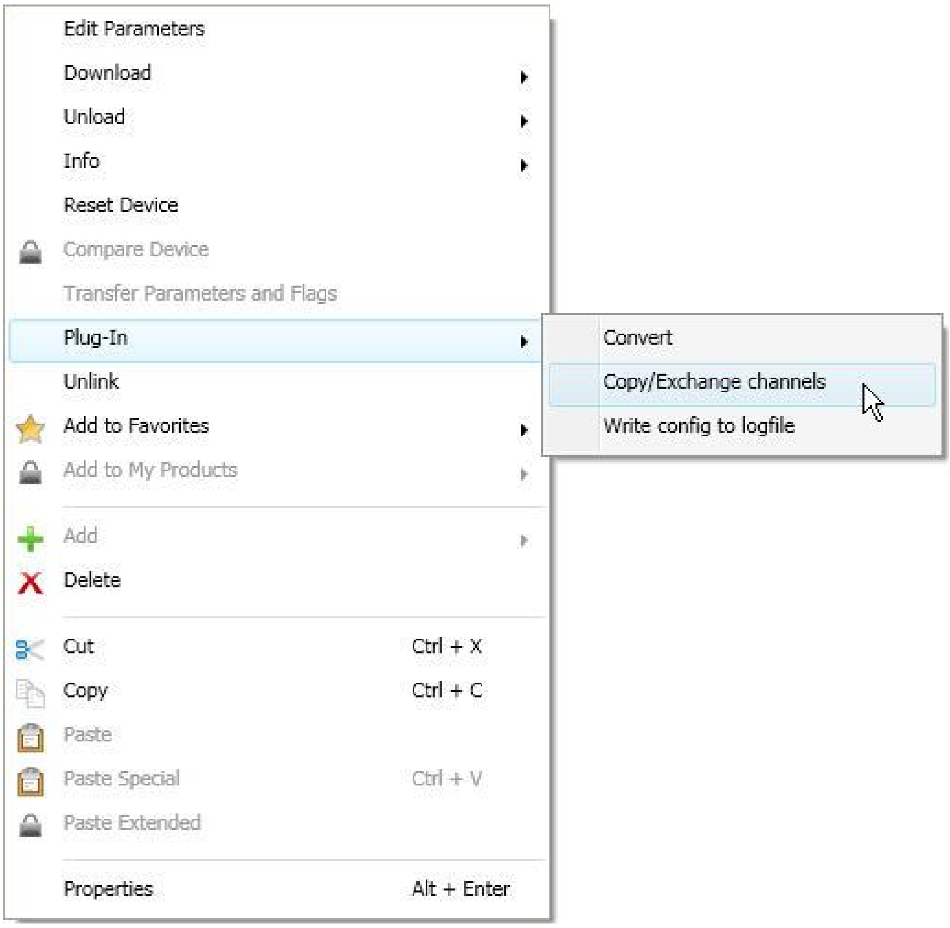
Note
When the term channels is used in the ETS, inputs and/or outputs are meant. In order to ensure that the ETS language generally applies for as many ABB i-bus® devices as possible, the word channels is used here.

The copy function for inputs/outputs is particularly useful with devices having the same parameter settings for several outputs, inputs or groups. For example, lighting in a room is frequently controlled in an identical manner. In this case, the parameter settings from input/output X can be copied to all other inputs/outputs or to a special input/output of the device. Thus the parameters for this input/output must not be set separately, which significantly shortens the commissioning time.

The exchange of parameter settings is useful, e.g. should the inputs/outputs be swapped when wiring the terminals. The parameter settings of the incorrectly wired inputs/outputs can be simply exchanged saving the requirement for time-consuming rewiring.

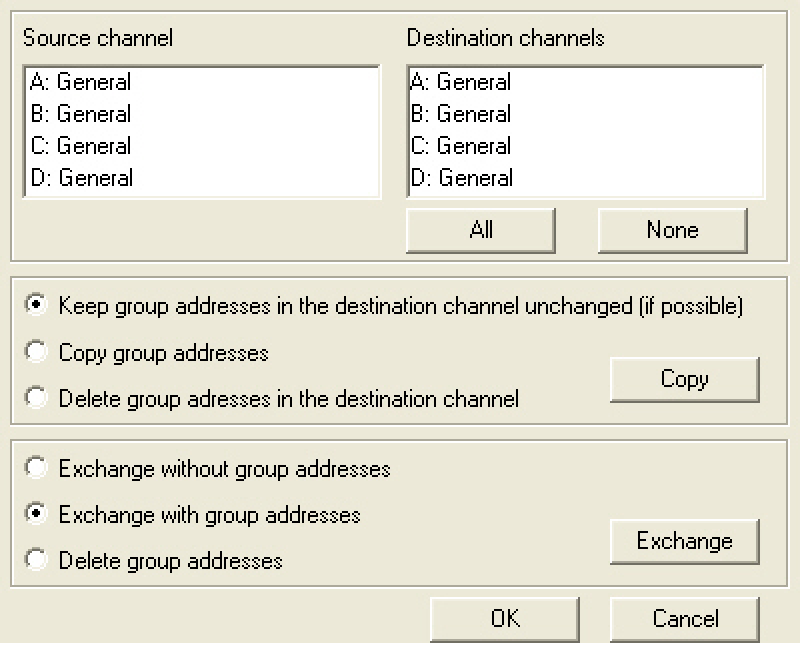
3.1.2.1 Procedure for copy and exchange

- Click with the right mouse button on the product, whose outputs you wish to copy or exchange, and select the context menu *Plug-in > Copy/exchange channels*.



Thereafter, undertake the required settings in the *Copy/exchange channels* dialog.

3.1.2.2 Dialog *Copy/exchange channel*



Source channel

A: General
B: General
C: General
D: General

Destination channels

A: General
B: General
C: General
D: General

● Keep group addresses in the destination channel unchanged (if possible)
○ Copy group addresses
○ Delete group adresses in the destination channel

○ Exchange without group addresses
● Exchange with group addresses
○ Delete group addresses

At the top left, you will see the Source channel selection window for marking the source channel. Beside is located the selection window for the target channel or channels for marking the target channel or channels.

Source channel

With the selection of the source channel, you define which parameter settings should be copied or exchanged. Only one source channel can be selected at a time.

Target channels

With the selection of the target channel/channels, you define which channel/channels are to assume the parameter settings of the source channel.

- For the function *Exchange*, only one target output can be selected at a time.
- For the function *Copy*, different target channels can be selected simultaneously. For this purpose, press the Ctrl key and mark the required channels with the mouse cursor, e.g. channels B and C.

All With this button, you select **all** available target channels, e.g. A…C.

None Reset the selection of the target channel with this button.

Copy

The following options can be selected before copying the parameter settings:

- Leave the group addresses unchanged (if possible) in the target channel
- Copy group addresses
- Delete group addresses in the target channel

With this button, copy the settings of the source channel into the target channel or channels.

Exchange

The following options can be selected before exchanging the parameter settings:

- Retain group addresses
- Exchange group addresses
- Delete group addresses

With this button, exchange the settings of the source channel with the target channel.

Confirm your selection with this button, and the window closes.

Using this button, the window closes without accepting the changes.

3.1.3 Functions of the outputs

The following table provides an overview of the functions possible with the outputs of the device and the application *Room Master*:

Functions of the outputs	**A…H**
Time	
Staircase lighting	■
ON/OFF delay	■
Flashing	■
Scene	
Assignment of the output to scenes	■
Logic	
AND/OR/XOR or GATE	■
Forced operation	
1 bit or 2 bit	■

■ = Function is supported

3.2 Parameters

The parameterization of the Room Master is implemented using the Engineering Tool Software ETS. The application is available in the ETS at *ABB/Room automation/Room Master*.

The following chapter describes the parameters of the device using the parameter windows. The parameter window features a dynamic structure, so that further parameters may be enabled depending on the parameterization and the function of the outputs.

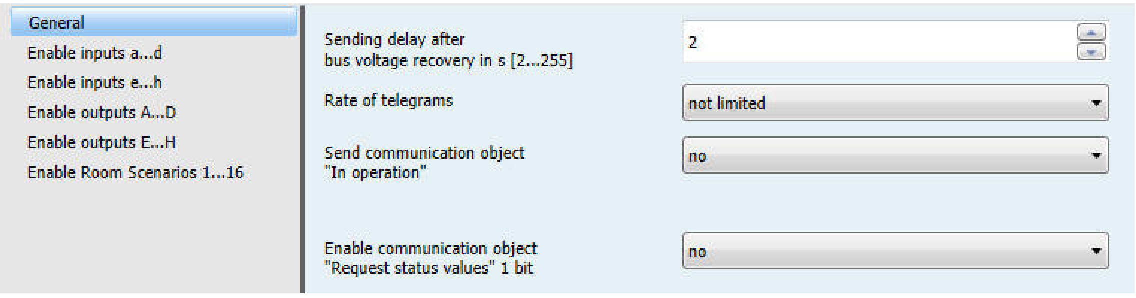
The default values of the parameters are underlined, e.g.:

Options: yes
<u>no</u>

Note
The device features several inputs/outputs. As the functions are identical for all inputs/outputs, they will only be explained using input/output A as an example.

3.2.1 Parameter window *General*

In this parameter window, higher level parameters can be set.



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

Rate of telegrams: not limited

Send communication object "In operation": no

Enable communication object "Request status values" 1 bit: no

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

Options: 2…255

Telegrams are only received during the sending and switching delay. The telegrams are not processed, however, and the outputs remain unchanged. No telegrams are sent on the bus.

After the sending and switching delay, telegrams are sent and the state of the outputs is set to correspond to the parameterization or the communication object values.

If communication objects are read during the sending and switching delay, e.g. by a visualisation system, these read requests are stored, and a response is sent, after the sending and switching delay has been completed.

An initialization time of about two seconds is included in the delay time. The initialisation time is the time that the processor requires to be ready to function.

How does the device behave with bus voltage recovery?

After bus voltage recovery, the device always waits for the send delay time to elapse before sending telegrams on the bus.

Rate of telegrams

Options: not limited
Send maximum 1 telegram/s
Send telegram every 0.1 s

- *Send maximum 1 telegram/s:* A maximum of one telegram per second is sent.
- *Send telegrams every 0.1 s:* A telegram is sent every 0.1 seconds.

This parameters limits the bus load of the device depending on its parameterization.

Send communication object
"In operation"

Options: no
send value 0 cyclically
send value 1 cyclically

The communication object In operation indicates the presence of the device on the bus. This cyclic telegram can be monitored by an external device. If a telegram is not received, the device may be defective or the bus cable to the transmitting device may be interrupted.

- *no:* The communication object *In operation* is not enabled.
- *send value 0/1 cyclically:* The communication object *In operation* (No. 0) is sent cyclically on the KNX. The following parameter appears:

Sending cycle time
in s [1...65,535]

Options: 1…60…65,535

Here the time interval, at which the communication object *In operation* (No. 0) cyclically sends a telegram, is set.

Note
After bus voltage recovery, the communication object sends its value after the set sending and switching delay.

Enable communication object
"Request status values" 1 bit

Options: no
yes

- *yes:* A 1 bit communication object *Request status values* is enabled.

Via this communication object, all status messages can be requested, provided that they have been parameterized with the option *after a change or request*.

With the option yes, the following parameters appear:

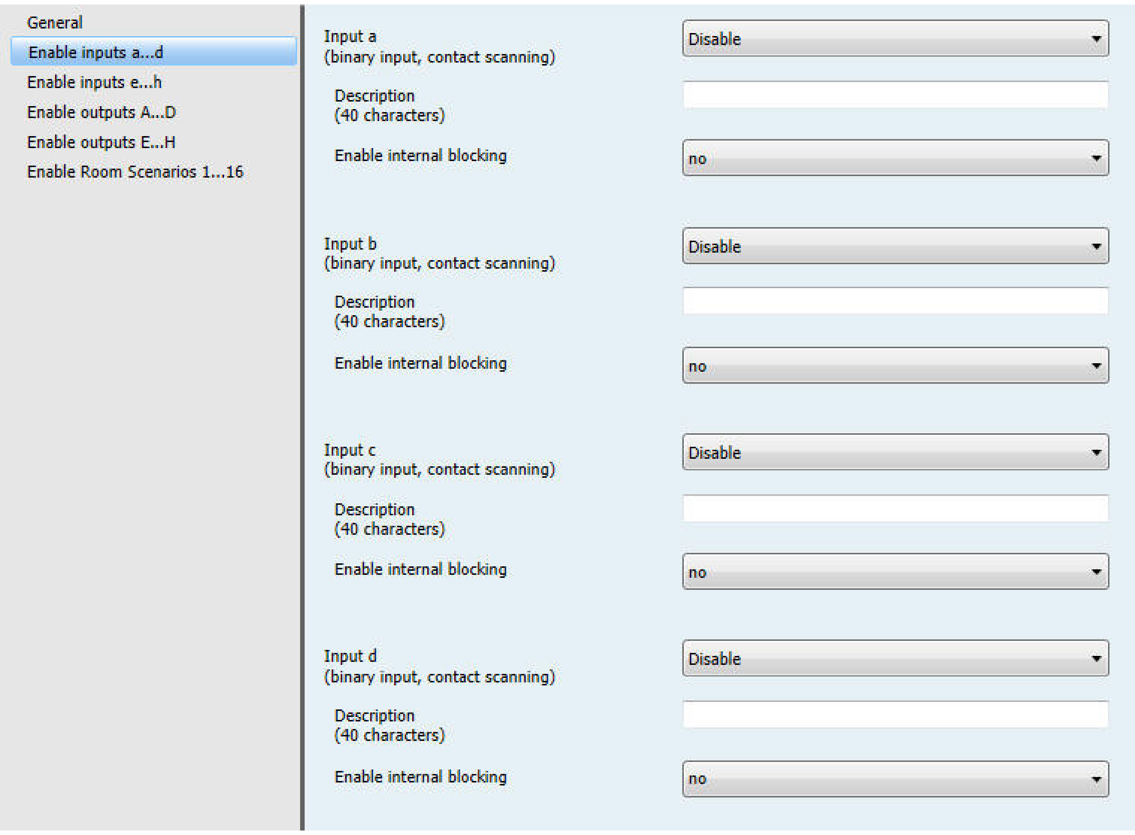
Request with object value

Options: 0
1
0 or 1

- *0:* Sending status messages is requested with the value 0.
- *1:* Sending status messages is requested with the value 1.
- *0 or 1:* Sending status messages is requested with the values 0 or 1.

3.2.2 Parameter window *Enable Inputs a…h*

In this parameter window, all the settings for enabling and description of the inputs a…h are undertaken.



Input a
(binary input, contact scanning) Disable

Description
(40 characters)

Enable internal blocking no

Input b
(binary input, contact scanning) Disable

Description
(40 characters)

Enable internal blocking no

Input c
(binary input, contact scanning) Disable

Description
(40 characters)

Enable internal blocking no

Input d
(binary input, contact scanning) Disable

Description
(40 characters)

Enable internal blocking no

Note

In the following, the setting possibilities of Inputs a…h are explained using input a as an example. The setting possibilities are identical for all inputs.

Input a
(binary input, contact scanning)

Option: Disabled
Switch sensor
Dim sensor
Blind sensor
Value / forced operation

The operating mode of the input is set with this parameter. The respective parameter window *a: xxx* also becomes visible with the selection of an operating mode.

Description
(40 characters)

With this parameter, it is possible to enter a text of up to 40 characters in length for identification in the ETS .

Note
The text which is entered is used to provide help, in order to obtain an overview of the inputs when they are fully assigned and to indicate the function assigned to the input. The text is purely for informative purposes and has no further function.

Enable internal blocking

Options: no
yes

This parameter defines whether a binary input can or cannot be internally inhibited. If an internal block is called, the binary input is physically disabled. Pressing a connected button/switch as well as incoming telegrams on communication object *Start event 0/1* are ignored.

This parameterization option enables the establishment of a blocking mask for all eight binary inputs. This blocking mask may also be called at every room state. It is thus possible to block (inhibit) or enable the binary inputs using this mask when this room state is recalled.

- *no:* The input cannot be blocked internally nor via the communication object *Block*.
- *yes:* The input can be blocked internally.

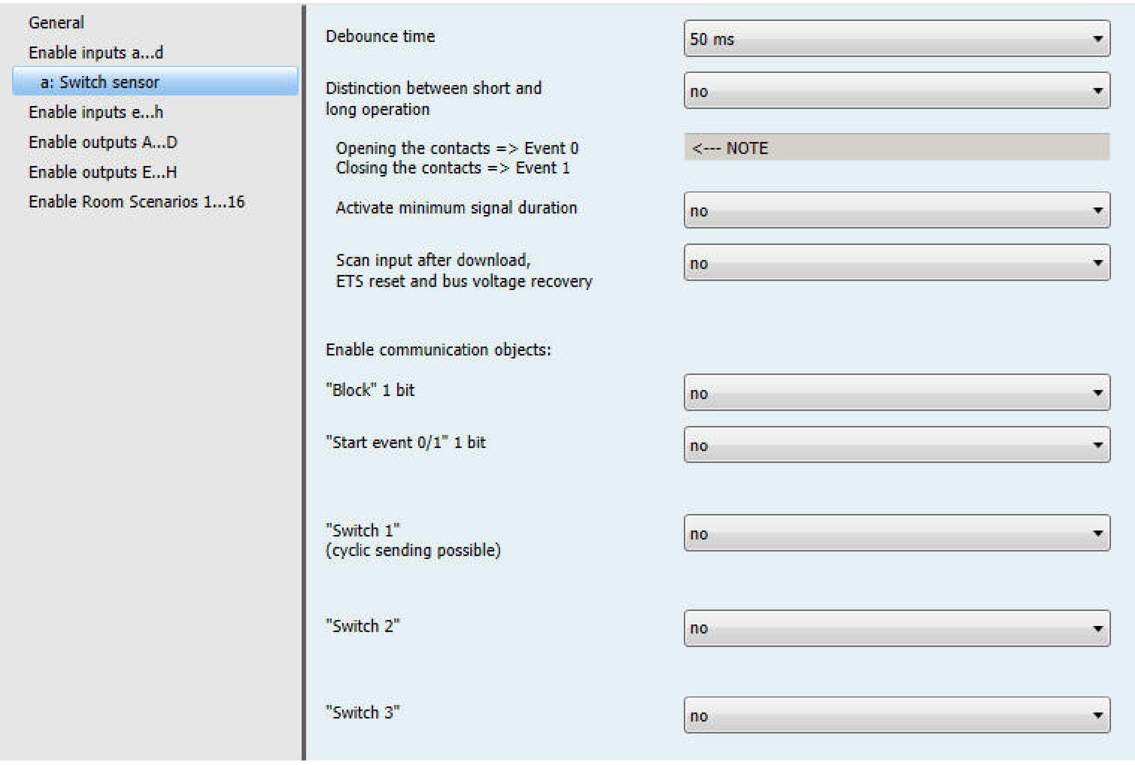
Inputs b…h

The device features several inputs. However, as the functions for all inputs are identical, only the functions of input a will be described.

3.2.2.1 Parameter window *a: Switch sensor*

This parameter window is visible if in Parameter window Enable Inputs a…h, page 26, in parameter *Input a (binary input, contact scanning)*, the option *Switch sensor* has been selected.

Note
The device features several inputs. However, as the functions for all inputs are identical, only the functions of input a will be described.



Debounce time: 50 ms

Distinction between short and long operation: no

Opening the contacts => Event 0
Closing the contacts => Event 1: <--- NOTE

Activate minimum signal duration: no

Scan input after download, ETS reset and bus voltage recovery: no

Enable communication objects:

"Block" 1 bit: no

"Start event 0/1" 1 bit: no

"Switch 1" (cyclic sending possible): no

"Switch 2": no

"Switch 3": no

Debounce time

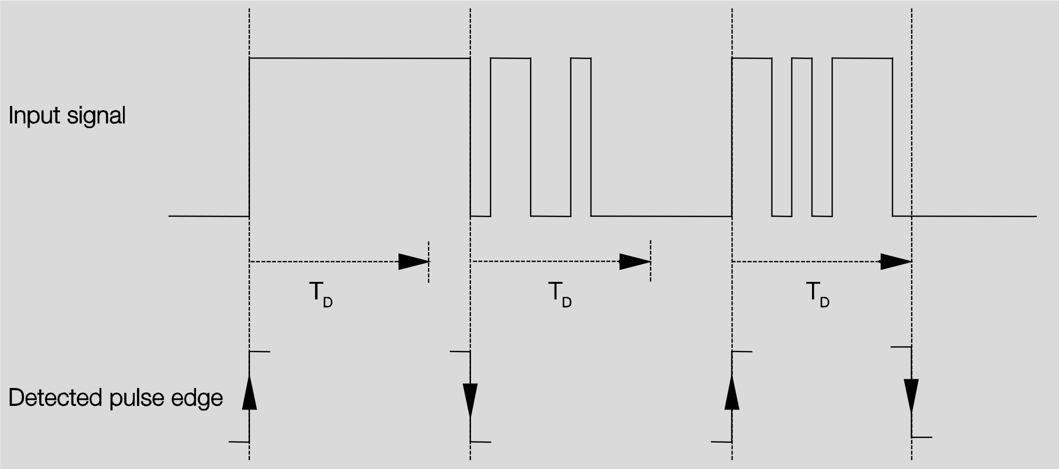
Options: 10/20/30/50/70/100/150 ms

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact.

What is the 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.

Example: Debounce time of the input signal for a detected edge:



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

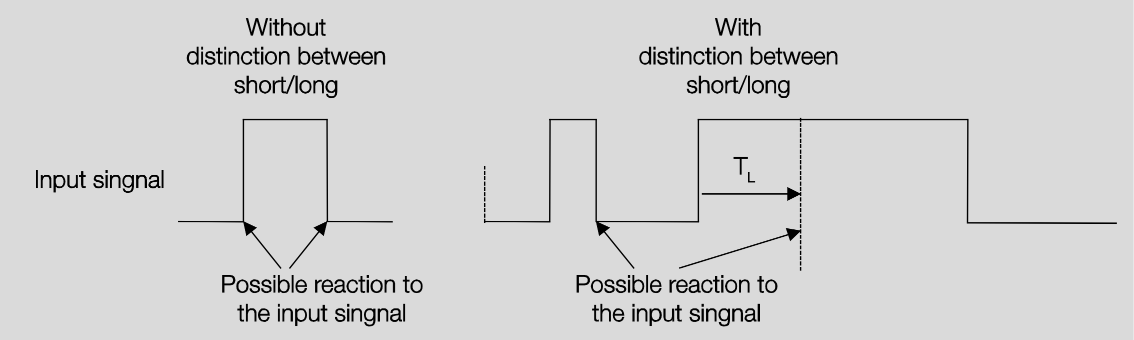
Distinction between short and long operation

Options: no
yes

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

- *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.

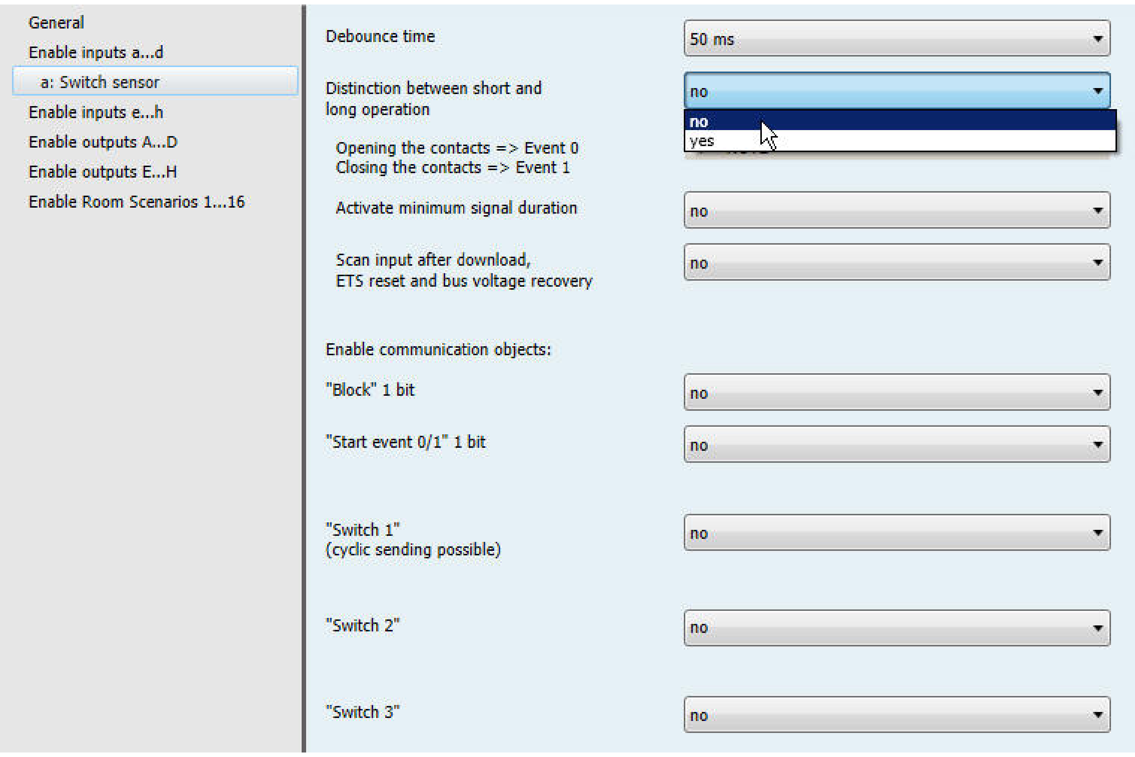
The following table shows the function in detail:



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

3.2.2.1.1 Parameter *Distinction between short and long operation – no*

If the option *no* is selected with the parameter *Distinction between short and long operation*, the following parameters appears in Parameter window a: Switch sensor, page 28:



Debounce time: 50 ms

Distinction between short and long operation: no

Opening the contacts => Event 0
Closing the contacts => Event 1

Activate minimum signal duration: no

Scan input after download, ETS reset and bus voltage recovery: no

Enable communication objects:

"Block" 1 bit: no

"Start event 0/1" 1 bit: no

"Switch 1" (cyclic sending possible): no

"Switch 2": no

"Switch 3": no

Opening the contacts => Event 0
Closing the contacts => Event 1

<--- NOTE

Activate minimum signal duration

Options: no
yes

- *yes:* The following parameters appear:

  On closing the contact
  in value x 0.1 s [0…65,535]

  Options: 1…10…65,535

  On opening the contact
  in value x 0.1 s [0…65,535]

  Options: 1…10…65,535

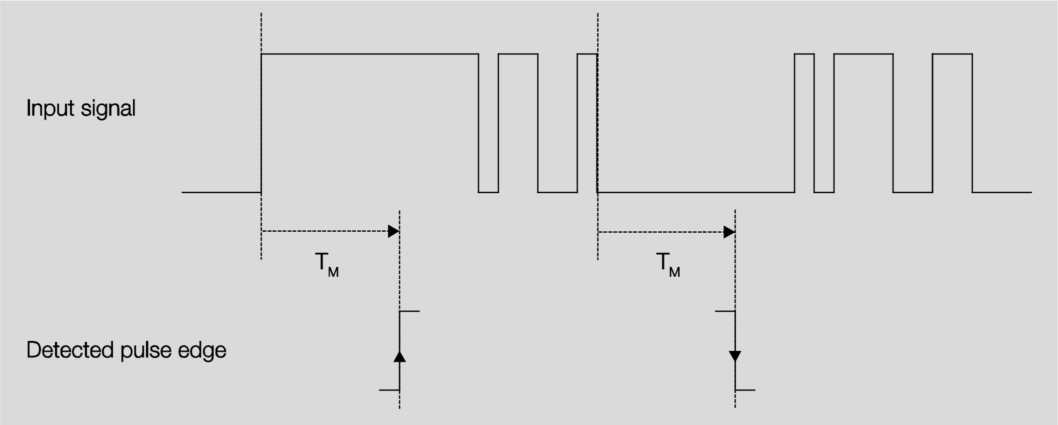
What is the minimum signal duration?

In contrast to the debounce time, a telegram is only sent after the minimum signal duration has elapsed.

The individual functions are:

If an edge is detected on the input, the minimum signal duration will commence. No telegram is sent on the bus at this time. The signal on the input is observed within the minimum signal duration. If a further edge appears at the input during the minimum signal duration, it will be interpreted as a new operation, and the minimum signal duration restarts. If no further edges occur after the start of the minimum signal duration, a telegram is sent on the bus, after the minimum signal duration has timed out.

Example: Minimum signal duration of the input signal for a detected edge:



In only two cases, no further edge changes occur within the minimum signal duration T_M after a change of edge. For this reason, only both of these are detected as valid.

Scan input after download,
ETS reset and bus voltage recovery

Options: <u>no</u>
yes

- *no:* The object value is not scanned after a download, bus reset and bus voltage recovery.
- *yes:* The object value is scanned after a download, bus reset and bus voltage recovery. The following parameter appears:

Inactive wait state after bus voltage recovery in s [0…30,000]

Options: 0…30,000

Here the waiting time after a bus voltage recovery is set. After the waiting time has elapsed the state on the input terminals is scanned. The input reacts as if the state on the input terminals has just changed.

Note
The inactive waiting time does not add to the actual, adjustable send delay time. This can be set separately.

Enable communication objects:

"Block" 1 bit

Options: no
yes

- *yes:* The 1 bit block communication object *Block* is enabled. This can be used to block the input.

Notes
If the input is disabled and the option *Cyclic sending* is set, the last state is still sent regardless of the block. The option *Block* still blocks the physical input, sending continues internally. Should the internal block with a binary input not be permitted in the Parameter window Enable Inputs a…h, page 26, this communication object has no effect on the respective binary input.

"Start event 0/1" 1 bit

Options: no
yes

- *yes:* The 1 bit communication object *Start event 0/1* is enabled. As a result, the same events, such as those of the push button/switch connected to the binary input, can also be triggered by the receipt of a telegram on the communication object *Start event 0/1*.

"Switch 1"
(cyclic sending possible)

Options: no
yes

- *yes:* The communication object *Switch 1* appears. The following parameters appear:

Reaction on event 0

Options:	ON OFF TOGGLE no reaction terminate cyclic transmission

Reaction on event 1

Options:	ON OFF TOGGLE no reaction terminate cyclic transmission

The behaviour of the communication object is determined here. If the option *yes* has been selected with the parameter *Distinction between short and long operation*, the reaction occurs with a short or long operation. With the option *no*, it occurs with each edge change.

Important
If the option *terminate cyclic sending* is set, it is important to note that this is only effective if the option *yes* has only been selected in the parameter *Cyclic sending*.

Internal connection

Options: no
Output A (6 A)
Output B (6 A)
Output C (6 A)
Output D (6 A)
Output E (6 A)
Output F (6 A)
Output G (6 A)
Output H (6 A)
Room Scenario 1/2
Room Scenario 3/4
Room Scenario 5/6
Room Scenario 7/8
Room Scenario 9/10
Room Scenario 11/12
Room Scenario 13/14
Room Scenario 15/16

With this parameter, a direct connection of the binary input with an output or with a Room Scenario can be established. With this connection, no assignment of the group address is necessary.

- *Output x:* The communication object *Switch* of the output is updated together with the communication object *Switch 1* of the binary input.

Caution

If an internal connection with an output is selected, and at the same time the reaction to an event is parameterized with TOGGLE, the communication object *Switch 1* of the binary input is updated with the inverted value of the communication object *Status Switch* of the output.

Ensure that the communication object *Status Switch* of the output is enabled. The settings *normally closed contact/normally open contact* and *Invert status* should be parameterized, so that a TOGGLE function is possible.

- *Room Scenario x/y:* If the communication object *Switch 1* is updated with the value 0, a Room Scenario (RS) with an odd number is triggered, i.e. RS 1/3/5/7/9/11/13 or 15. If the communication object *Switch 1* is updated with the value 1, a Room Scenario (RS) with an even number is triggered, i.e. RS 2/4/6/8/10/12/14 or 16.

Cyclic sending

Options: <u>no</u>
yes

What is cyclic sending?

Cyclic sending enables the communication object *Switch* to send automatically 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 behaviour is unwanted, the flags *Write* and *Update* of the communication object are deleted in the preliminary setting, so that they cannot be changed via the bus. If this functionality is required irrespectively, these flags should be set accordingly. When the communication object *Switch* changes and after bus recovery (after the send delay time has elapsed), the communication object value is sent immediately on the bus, and the transmission cycle time restarts.

- *yes:* The following parameters appear:

**Telegram repeated every …
in s [1...65,535]**

Options: 1…<u>60</u>…65,535

The send cycle time describes the time used between two cyclically sent telegrams.

On object value

Options: 1
0
<u>0 or 1</u>

- *1:* The communication object value is sent cyclically with 1.
- *0:* The communication object value is sent cyclically with 0.
- *0 or 1:* The communication object values 0 and 1 are sent cyclically.

"Switch 2"

"Switch 3"

Options: no
Yes

- *yes:* The communication object *Switch 2/3* becomes visible. The following parameters appear:

Reaction on event 0

Options: ON
OFF
TOGGLE
no reaction

Reaction on event 1

Options: ON
OFF
TOGGLE
no reaction

The behaviour of the communication object is determined here. If the option *yes* has been selected with the parameter *Distinction between short and long operation*, the reaction occurs with a short or long operation. With the option *no*, it occurs with each edge change.

Internal connection

Options: no
Output A (6 A)
Output B (6 A)
Output C (6 A)
Output D (6 A)
Output E (6 A)
Output F (6 A)
Output G (6 A)
Output H (6 A)
Room Scenario 1/2
Room Scenario 3/4
Room Scenario 5/6
Room Scenario 7/8
Room Scenario 9/10
Room Scenario 11/12
Room Scenario 13/14
Room Scenario 15/16

With this parameter, a direct connection of the binary input with an output or with a Room Scenario can be established. With this connection, no assignment of the group address is necessary.

- *Output x:* The communication object *Switch* of the output is updated together with the communication object *Switch 2/3* of the binary input.

Caution

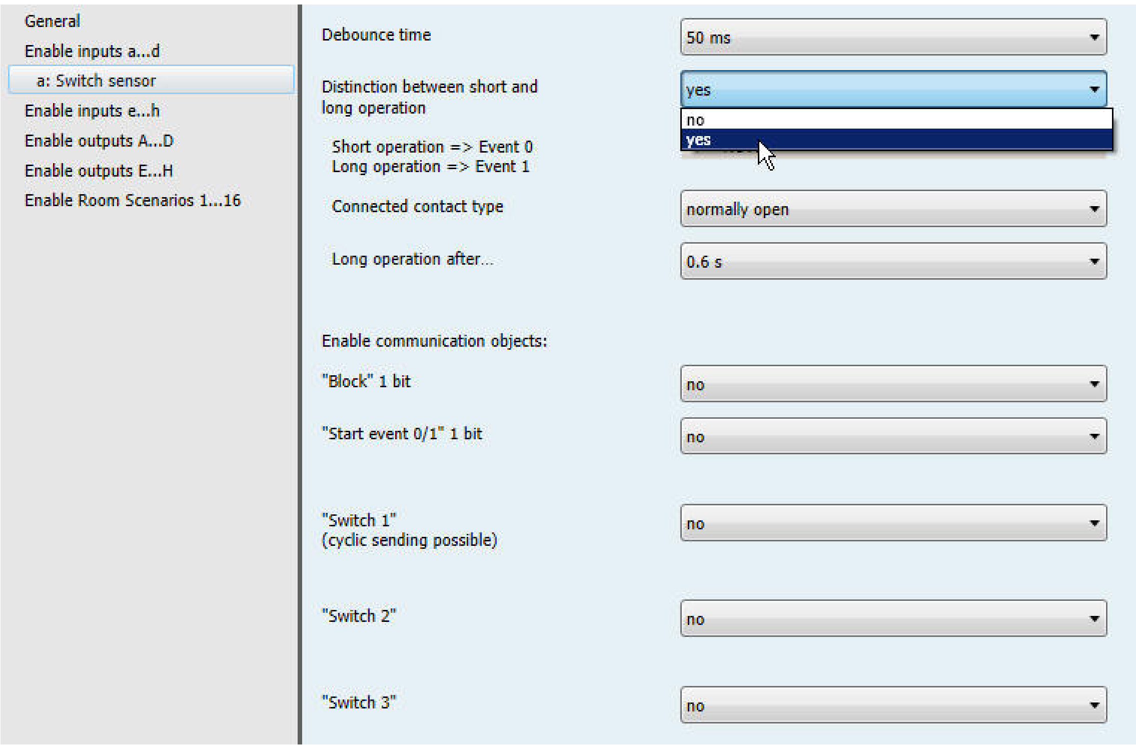
If an internal connection with an output is selected, and at the same time the reaction to an event is parameterized with TOGGLE, the communication object *Switch 2/3* of the binary input is updated with the inverted value of the communication object *Status switch* of the output.

Ensure that the communication object *Status Switch* of the output is enabled. The settings *normally closed contact/normally open contact* and *Invert status* should be parameterized, so that a TOGGLE function is possible.

- *Room Scenario x/y:* If the communication object *Switch 2/3* is updated with the value 0, a Room Scenario (RS) with an odd number is triggered, i.e. RS 1/3/5/7/9/11/13 or 15. If the communication object *Switch 2/3* is updated with the value 1, a Room Scenario (RS) with an even number is triggered, i.e. RS 2/4/6/8/10/12/14 or 16.

3.2.2.1.2 Parameter *Distinction between short and long operation – yes*

If the option *yes* is selected with the parameter *Difference between long and short operation*, the following parameters are visible in Parameter window a: Switch sensor on page 28.



Debounce time	50 ms
Distinction between short and long operation	yes
Short operation => Event 0 Long operation => Event 1	
Connected contact type	normally open
Long operation after...	0.6 s
Enable communication objects:	
"Block" 1 bit	no
"Start event 0/1" 1 bit	no
"Switch 1" (cyclic sending possible)	no
"Switch 2"	no
"Switch 3"	no

Short operation => Event 0
Long operation => Event 1

<––– NOTE

Connected contact type

Options: normally closed
normally open

- *normally closed:* The input is opened on actuation.
- *normally open:* The input is closed on actuation.

If a normally open contact is connected to the input, the option *normally open* should be selected; on a normally closed contact the option *normally closed* should be selected.

Long operation after…

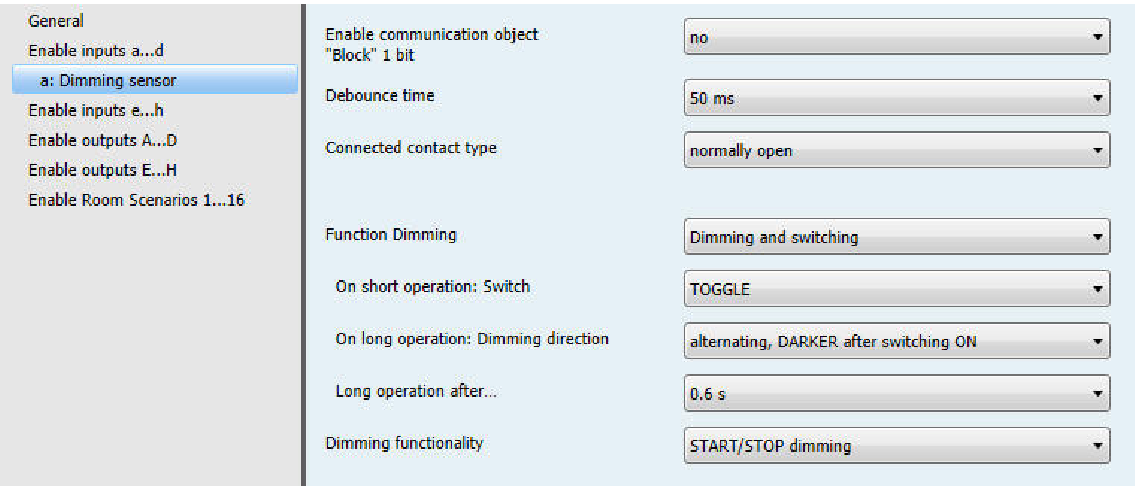
Options: 0.3/0.4/0.5/0.6/0.8 s
1/1.2/1.5 s
2/3/4/5/6/7/8/9/10 s

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

Note
The remaining parameter descriptions can be found in the Parameter Distinction between short and long operation – no, page 30.

3.2.2.2 Parameter window *a: Dimming sensor*

The operating mode allows the operation of dimmable lighting. This parameter window is visible if in Parameter window Enable Inputs a…h, page 26, in parameter *Input a (binary input, contact scanning)*, the option *Dimming sensor* has been selected.



Enable communication object "Block" 1 bit: no

Debounce time: 50 ms

Connected contact type: normally open

Function Dimming: Dimming and switching

On short operation: Switch: TOGGLE

On long operation: Dimming direction: alternating, DARKER after switching ON

Long operation after...: 0.6 s

Dimming functionality: START/STOP dimming

Enable communication object "Block" 1 bit

Options: no
yes

- *yes:* The 1 bit block communication object *Block* is enabled. This can be used to block the input.

Note
If the input is disabled and the option *Cyclic sending* is set, the last state is still sent regardless of the block. The option *Block* still blocks the physical input, sending continues internally.

Debounce time

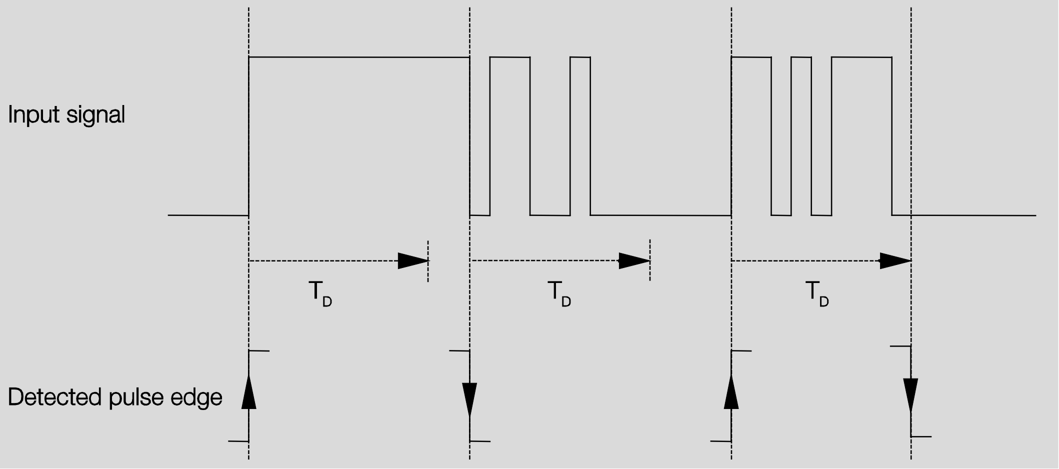
Options: 10/20/30/50/70/100/150 ms

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact.

What is the 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.

Connected contact type

Options: normally closed
normally open

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

Function Dimming

Options: Dimming and switching
Only dimming

With this parameter, you 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.

How does 1 button dimming function?

Switch and dim functions can be controlled completely using a single push button. With each long operation alternate BRIGHTER or DARKER dimming occurs, or with short operation alternate switch on or off occurs.

If the communication object *Switch* = 0, a BRIGHTER telegram is sent at all times. In order to evaluate the switch feedback of the actuator, the Write flag of the communication object *Switch* is set.

The following table shows the function in detail:

Communication object value ***Switch***	**Value of the last dimming telegram**	**Reaction of the dimming actuation (sends dimming telegram)**
OFF	DARKER	BRIGHTER
OFF	BRIGHTER	BRIGHTER
ON	DARKER	BRIGHTER
ON	BRIGHTER	DARKER

The advantage of the *Only dimming* function is that no distinction is made between short and long actuation. The dim telegram is initiated immediately after actuation in this way. It is not necessary to wait for a long operation.

How does 2 button dimming function?

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 thus has the choice of the buttons to be combined with one another, e.g. to dim a lighting group or the function that the individual buttons should perform in this case.

Furthermore, two inputs are required for 2 button dimming, e.g. *Input* a with short operation with switch ON and long operation for BRIGHTER dimming. *Input b* with short operation for switch OFF and long operation for DARKER dimming.

If the option *Dimming and switching* is selected with the parameter *Function Dimming*, the parameters *Long operation after…*, *On short operation*: *Switch* and *On long operation: Dimming direction* become visible in the parameter window *a: Dimming sensor:*

Long operation after…

Options: 0.3/0.4/0.5/<u>0.6</u>/0.8/1/1.2/1.5/2/3/4/5/6/7/8/9/10 s

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

On short operation: Switch

Options: ON
OFF
TOGGLE
no reaction

This parameter defines if the communication object *Telegram switch TOGGLEs* with short operation (typical: 1 button dimming) or only switches *OFF* or *ON* (typically: 2 button dimming).

- *ON:* With short operation the value 1 is sent.
- *OFF:* With short operation the value 0 is sent.
- *TOGGLE:* A short operation changes the value of the communication object *Telegram switch*.

On long operation: Dimming direction

Options: BRIGHTER
DARKER
alternating
alternating, BRIGHTER after switching ON
alternating, DARKER after switching ON

With this parameter, you set what the communication object *Dimming* should send on the bus with a long operation. A long operation changes the value of the communication object *Telegr. dimming.* With 1 button dimming, the parameter *Dimming* should be set with the option *alternating*. In this case, the dimming telegram, which is diametrically opposed to the last dimming telegram, is sent.

- *BRIGHTER:* The communication object sends a BRIGHTER telegram.
- *DARKER:* The communication object sends a DARKER telegram.
- *alternating:* The communication object alternately sends a BRIGHTER and a DARKER telegram.
- *alternating, BRIGHTER after switching ON*: The communication object at the first time sends a BRIGHTER telegram after an ON telegram; thereafter it alternately sends BRIGHTER and DARKER telegrams.
- *alternating, DARKER after switching ON:* The communication object at the first time sends a DARKER telegram after an ON telegram, thereafter it alternately sends BRIGHTER and DARKER telegrams.

Note
If the option *Only dimming* is selected in the *Function Dimming*, only the parameter *On operation: Dimming direction* is visible.

Dimming functionality

Options: START/STOP dimming
Dimming steps

- *START/STOP dimming:* The dimming process starts with a telegram BRIGHTER or DARKER and ends with a STOP telegram.

4 bit dimming telegram:

Decimal	Hexadecimal	Binary	Dim telegram
0	0	0000	STOP
1	1	0001	100 % DARKER
8	8	1000	STOP
9	9	1001	100 % BRIGHTER

For further information see: Input 4 bit dimming telegram, page 125.

- *Dimming steps:* Dimming telegrams are sent cyclically during a long operation. Cyclic sending is terminated after the end of actuation.

Both of the next parameters only appear if in the parameter *Dimming functionality* the option *Dimming steps* has been set.

Brightness change on every sent telegram

Options: 100/50/25/12.5/6.25/3.13/1.56 %

Using this parameter, you set the brightness change in percent which is cyclically sent with every dim telegram.

Sending cycle time: Telegram is repeated every …

Options: 0.3/0.4/0.5/0.6/0.8/1/1.2/1.5/2/3/4/5/6/7/8/9/10 s

The dimming telegram is sent cyclically during a long operation. The cycle time for sending corresponds with the time interval between two telegrams during cyclical sending.

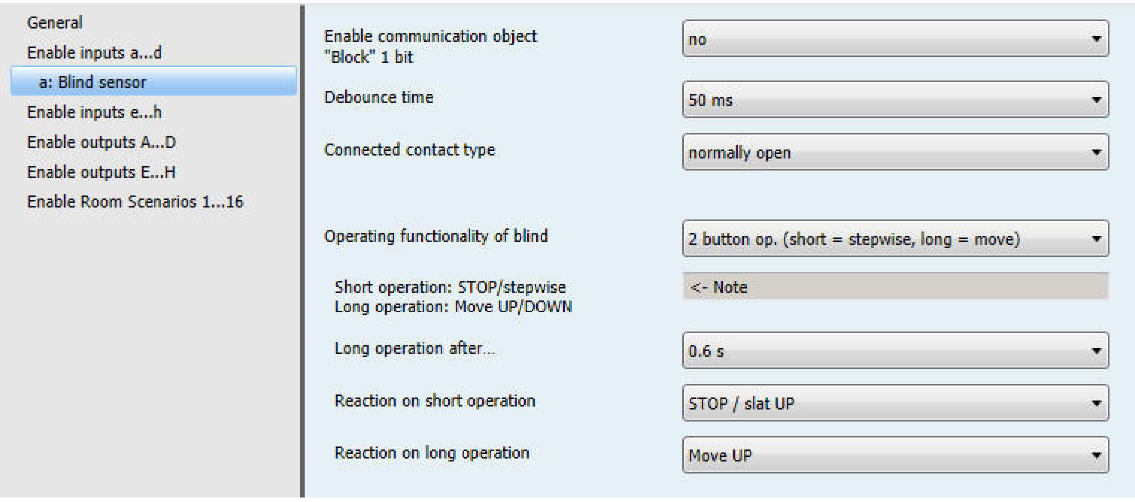
Caution

With dimming steps, you ensure that the set *Sending cycle time* is matched on the dimming actuator in order to enable a smooth dimming process.

3.2.2.3 Parameter window *a: Blind sensor*

The operating mode allows the operation of blinds and roller shutters with buttons or switches.

This parameter window is visible if in Parameter window Enable Inputs a…h, page 26, in parameter *Input a (binary input, contact scanning)*, the option *Blind sensor* has been selected.



- General
- Enable inputs a...d
  - a: Blind sensor
- Enable inputs e...h
- Enable outputs A...D
- Enable outputs E...H
- Enable Room Scenarios 1...16

Enable communication object "Block" 1 bit	no
Debounce time	50 ms
Connected contact type	normally open
Operating functionality of blind	2 button op. (short = stepwise, long = move)
Short operation: STOP/stepwise Long operation: Move UP/DOWN	<- Note
Long operation after...	0.6 s
Reaction on short operation	STOP / slat UP
Reaction on long operation	Move UP

Enable communication object "Block" 1 bit

Options: no
yes

- *yes:* The 1 bit block communication object *Block* is enabled. This can be used to block the input.

Note
If the input is disabled and the option *Cyclic sending* is set, the last state is still sent regardless of the block. The option *Block* still blocks the physical input, sending continues internally.

Debounce time

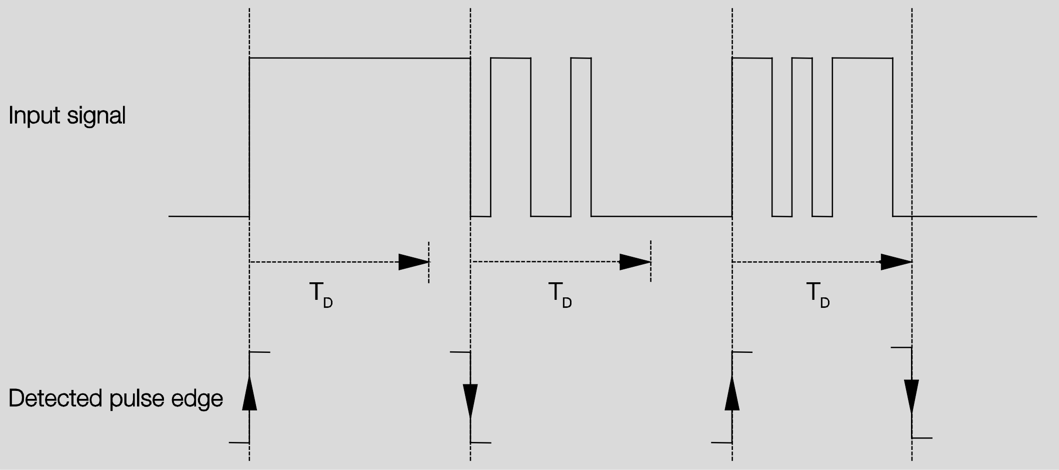
Options: 10/20/30/50/70/100/150 ms

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact.

What is the 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.

Connected contact type

Options: normally closed
<u>normally open</u>

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

Operating functionality of blind

Options:
1 button op. (short = stepwise, long = move)
1 button op. (short = move, long = stepwise)
1 push button (Move only - STOP)
1 switch operation (Move only)
<u>2 button op. (short = stepwise, long = move)</u>
2 switch/push button operation (moving only)
2 push button operation (move only)
2 push button operation (slat only)

The following list provides an overview of the different blind operating functions:

1 push buttons (short = stepwise, long = move)	
Short operation	STOP/Stepwise Opposite direction to the last movement telegram* To return to slat adjustment, the blind must be moved UP or DOWN briefly.
Long operation	*Move UP* or *Move DOWN*
1 push button op. (short = move, long = stepwise)	
Short operation	*Move UP* or *Move DOWN*
Long operation	STOP/stepwise (Cyclic sending); Opposite direction to the last movement telegram
1 push button (Move only - STOP)	
On operation	The following telegrams are sent in sequence: …► *Move UP* ► *STOP/Stepwise* ► *Move DOWN* ► *STOP/Stepwise* ► … *
1 switch operation (Move only)	
On operation	*Move UP* or *Move DOWN*
End of operation	STOP/Stepwise*
2 push button operation (short = stepwise, long = move)	
Short operation	*STOP/Slat OPEN/CLOSE* (programmable)
Long operation	*Move UP* or *Move DOWN* (programmable)
2 switch/push button operation (moving only)	
On operation	*Move UP* or *Move DOWN* (programmable)
End of operation	*STOP/Slat OPEN/CLOSE* (programmable)
2 push button operation (move only)	
On operation	*Move UP* or *Move DOWN* (programmable)
2 push button operation (slat only)	
On operation	*STOP/Slat OPEN/CLOSE* (programmable)

* If the actuator indicates the limit position, in 1 button operation the communication object *Blind UP/DOWN* can be synchronized. If the actuator signals the upper limit position (see communication object *Upper limit position* or *Lower limit position*), the direction of movement is defined. In 1 push button/switch operation, the last direction of movement is determined via the last update of the communication object *Blind UP/DOWN*.

Depending on the selection made in the parameter *Operating functionality of the blind*, different parameters will appear.

All parameters are described in the following.

Long operation after…

Options: 0.3/0.4/0.5/0.6/0.8/1/1.2/1.5/2/3/4/5/6/7/8/9/10 s

Here the time period T_L after which an operation is considered a "long" operation is defined.

Telegram "Slat" is repeated every …

Options: 0.3/0.4/0.5/0.6/0.8/1/1.2/1.5/2/3/4/5/6/7/8/9/10 s

The time duration, at which the telegram *Slat* is repeated, is defined here.

Reaction on short operation

Options: STOP/Slat OPEN
STOP/Slat CLOSE

Reaction on long operation

Options: Move UP
MOVE DOWN

It can be set whether the input triggers telegrams for movement upwards (UP) or downwards (DOWN).

Reaction on operation

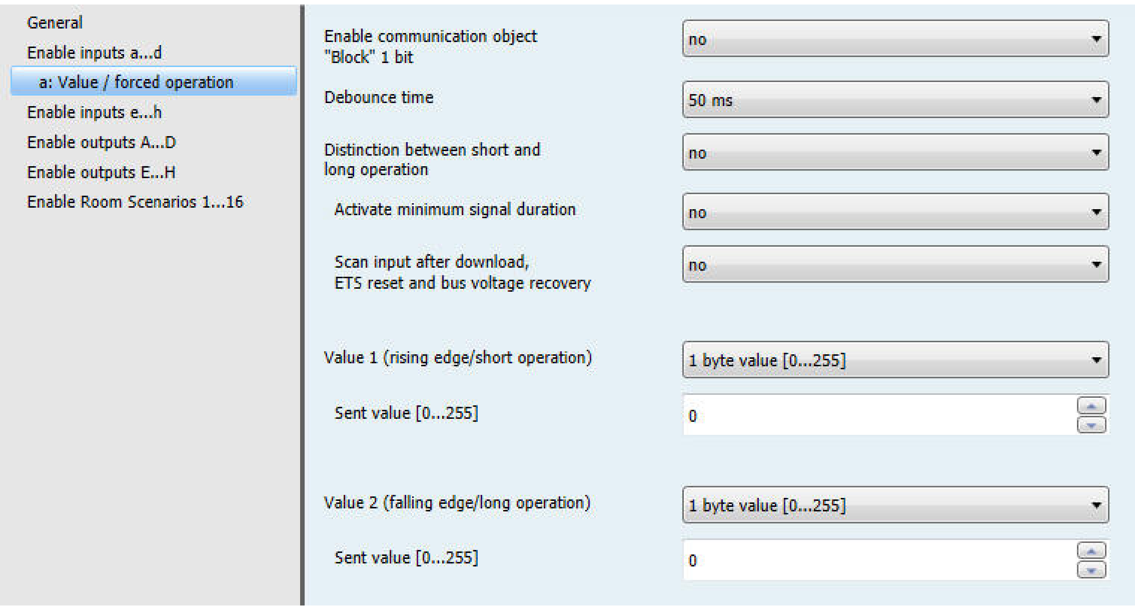
Options: Move UP
MOVE DOWN

It can be set whether the input triggers telegrams for movement upwards (UP) or downwards (DOWN).

3.2.2.4 Parameter window *a: Value/Forced operation*

This operating mode allows the sending of values of any data types.

This parameter window is visible if in Parameter window Enable Inputs a…h, page 26, in parameter *Input a (binary input, contact scanning)*, the option *Value/Forced operation* has been selected.



Enable communication object "Block" 1 bit: no
Debounce time: 50 ms
Distinction between short and long operation: no
Activate minimum signal duration: no
Scan input after download, ETS reset and bus voltage recovery: no
Value 1 (rising edge/short operation): 1 byte value [0...255]
Sent value [0...255]: 0
Value 2 (falling edge/long operation): 1 byte value [0...255]
Sent value [0...255]: 0

Enable communication object
"Block" 1 bit

Options: <u>no</u>
yes

- *yes:* The 1 bit block communication object *Block* is enabled. This can be used to block the input.

Note
If the input is disabled and the option *Cyclic sending* is set, the last state is still sent regardless of the block. The option *Block* still blocks the physical input, sending continues internally.

Debounce time

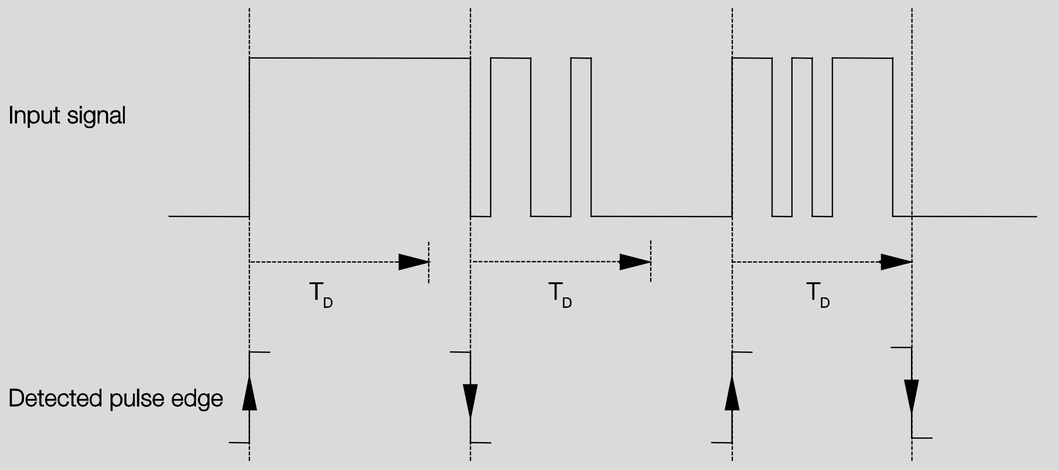
Options: 10/20/30/<u>50</u>/70/100/150 ms

Debouncing prevents unwanted multiple operations of the input, e.g. due to bouncing of the contact.

What is the 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.

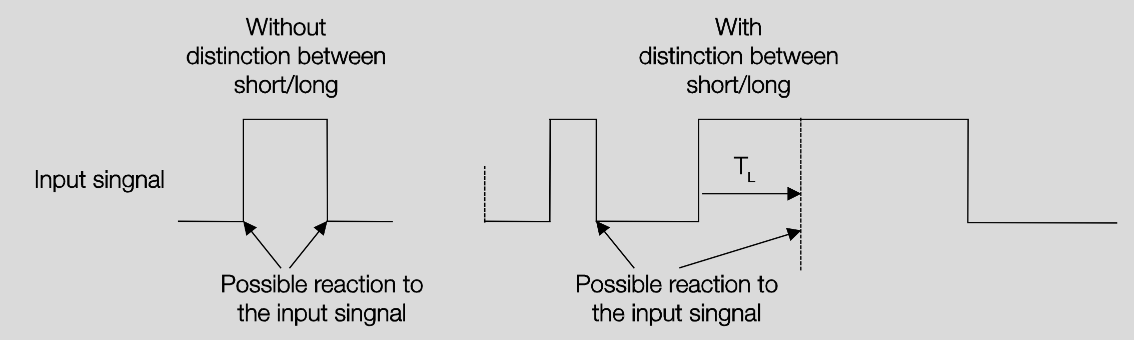
Distinction between short and long operation

Options: no
yes

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

Note
With *Distinction between short and long operation*, two communication objects are visible for each input. One communication object only transmits during short operation, the other communication object only during a long operation.

The following table shows the function in detail:

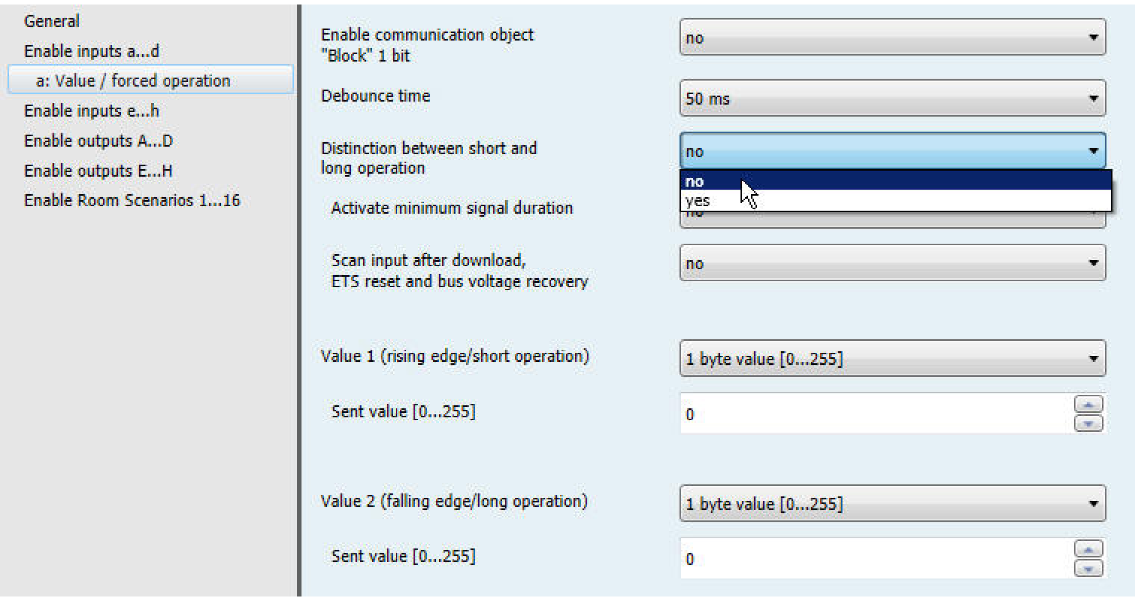


T_L is the time duration from where a long operation is detected.

If the option *no* is selected with the parameter *Distinction between short and long operation*, the following parameters appear:

3.2.2.4.1 Parameter *Distinction between short and long operation – no*

If the option *no* is selected with the parameter *Distinction between short and long operation*, the following parameters appears in Parameter window a: Value/Forced operation, page 48:



Enable communication object "Block" 1 bit: no

Debounce time: 50 ms

Distinction between short and long operation: no (no / yes)

Activate minimum signal duration: no

Scan input after download, ETS reset and bus voltage recovery: no

Value 1 (rising edge/short operation): 1 byte value [0...255]

Sent value [0...255]: 0

Value 2 (falling edge/long operation): 1 byte value [0...255]

Sent value [0...255]: 0

Activate minimum signal duration

Options: <u>no</u>
yes

- *yes:* The following parameters appear:

for rising edge
in value x 0.1 s [1…65,535]

Options: 1…<u>10</u>…65,535

Note
A rising edge corresponds to a “normally opened contact function”.

for falling edge
in value x 0.1 s [1…65,535]

Options: 1…<u>10</u>…65,535

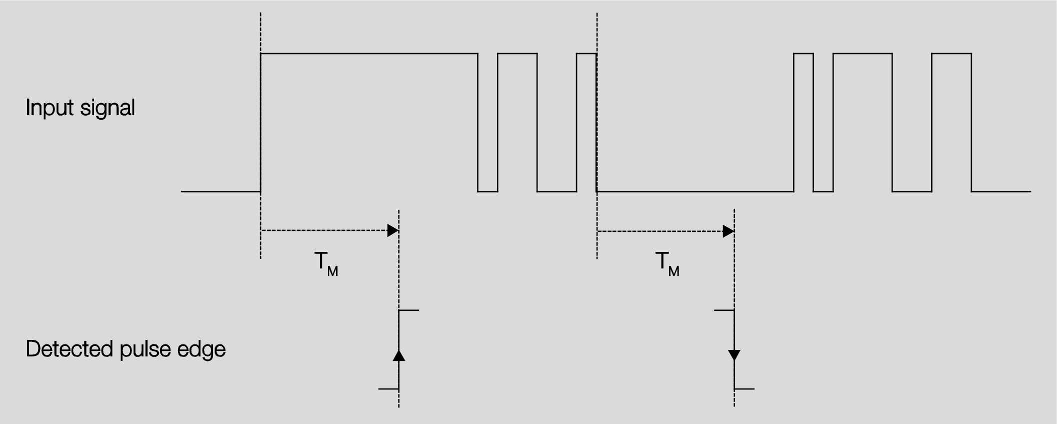
Note
A falling edge corresponds to a normally closed contact function.

What is the minimum signal duration?

In contrast to the debounce time, a telegram is only sent after the minimum signal duration has elapsed. The individual functions are:

If an edge is detected on the input, the minimum signal duration will commence. No telegram is sent on the bus at this time. The signal on the input is observed within the minimum signal duration. If a further edge appears at the input during the minimum signal duration, it will be interpreted as a new operation, and the minimum signal duration restarts. If no further edges occur after the start of the minimum signal duration, a telegram is sent on the bus, after the minimum signal duration has timed out.

Example: Minimum signal duration of the input signal for a detected edge:



In only two cases, no further edge changes occur within the minimum signal duration T_M after a change of edge. For this reason, only both of these are detected as valid.

Scan input after download,
ETS reset and bus voltage recovery

Options: no
yes

- *no:* The object value is not scanned after a download, bus reset and bus voltage recovery.
- *yes:* The object value is scanned after a download, bus reset and bus voltage recovery. The following parameter appears:

Inactive wait state after bus
voltage recovery in s [0…30,000]

Options: 0…30,000

Here the waiting time after a bus voltage recovery is set. After the waiting time has elapsed the state on the input terminals is scanned. The input reacts as if the state on the input terminals has just changed.

Note
The inactive waiting time does not add to the actual, adjustable send delay time. This can be set separately.

Value 1 (rising edge/short operation)

Options: do not send
1 bit value [0/1]
2 bit value [Forced operation]
1 byte value [-128…127]
1 byte value [0…255]
1 byte value [8 bit scene]
2 byte value [-32,768...32,767]
2 byte value [0...65,565]
2 byte value [floating point]
3 byte value [time of day, weekday]
4 byte value [-2147483648…2147483647]
4 byte value [0…4294967295]

This parameter serves for defining the data type which is sent when the contact is actuated.

Depending on the selection made in parameter *Value 1 (rising edge / short operation)*, different parameters appear. All parameters are described in the following:

Sent value [X]

Options: ON /OFF/ TOGGLE
0/1
-128…0…127
0…255
-32. 768…0…32. 767
0…65,535
-100…20…100
-2147483648…0…2147483647
0…4294967295

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

Sent value

Options: ON, activate forced position
OFF, activate forced operation
Disable forced operation

This parameter defines the value which is sent on operation.

In the following table, the *Forced operation* function is explained:

Bit 1	Bit 0	Access	Description
0	0	Free	The switch communication object of the actuator is enabled by the binary input. The assigned sensor can control the actuator via the switch object. The binary input does not control the actuator. Bit 0 of the value of the forced operation communication object is not evaluated. The forced operation communication object sends a telegram with the group addresses of the forced operation communication object and the status of the switch communication object with every state change of the switch communication object.
0	1	Free	
1	0	OFF	The switch communication object of the actuator is disabled by the binary input. The assigned sensor can not control the actuator via the switch communication object. The binary input controls the actuator via the forced operation communication object. The actuator is switched off. Bit 0 of the value of the forced operation communication object is evaluated.
1	1	ON	The switch communication object of the actuator is disabled by the binary input. The assigned sensor can not control the actuator via the switch communication object. The binary input controls the actuator via the forced operation communication object. The actuator is switched ON.

8 bit scene

Options: 1…64

This parameter defines the scene number, which is sent on actuation.

Recall/store scene

Options: recall
Save

This parameter defines whether the scene is to be recalled or stored.

Hour [0...23]

Options: 0…23

Minute [0...59]

Options: 0…59

Seconds [0…59]

Options: 0…59

With these parameters, the hours, minutes and seconds are set which are to be send when actuated.

Weekday [1 = Mo, 2..6, 7 = Su]

Options: <u>0 = no day</u>
1 = Monday
2 = Tuesday
3 = Wednesday
4 = Thursday
5 = Friday
6 = Saturday
7 = Sunday

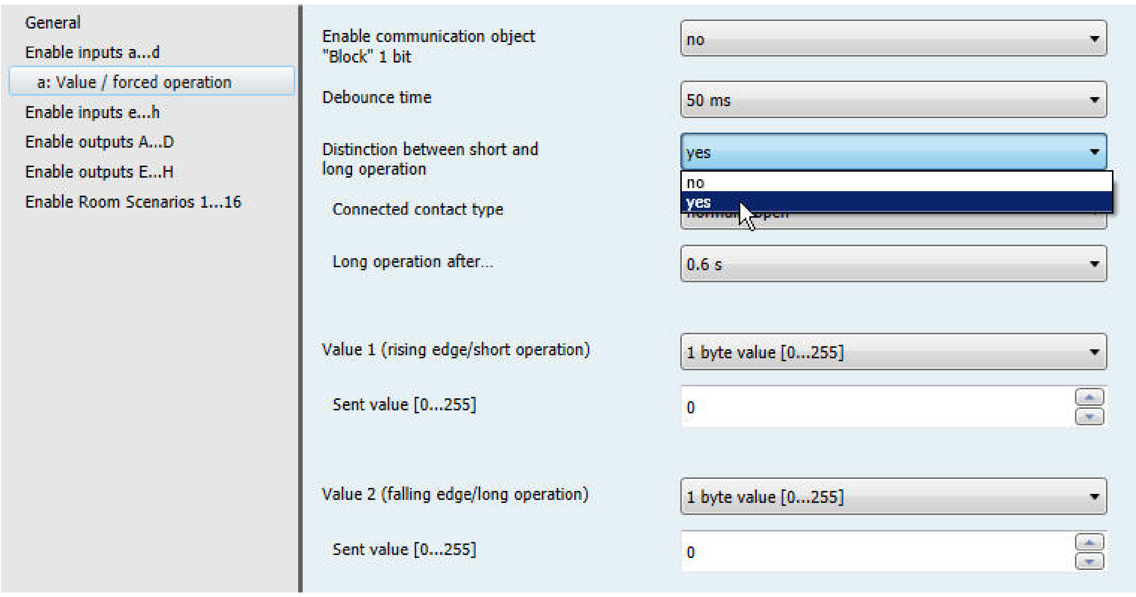
Using these parameters, the weekdays sent on actuation are set.

Value 2 (falling edge/long operation)

Note
The parameter descriptions of the parameter *Value 2 (with a rising edge and with short operation)* correspond with those of parameters *Value 1 (with a rising edge and with short operation)*.

3.2.2.4.2 Parameter *Distinction between short and long operation – yes*

If the option *yes* is selected with the parameter *Distinction between short and long operation*, the following parameters appear:



Enable communication object "Block" 1 bit: no

Debounce time: 50 ms

Distinction between short and long operation: yes

Connected contact type: normally open

Long operation after...: 0.6 s

Value 1 (rising edge/short operation): 1 byte value [0...255]

Sent value [0...255]: 0

Value 2 (falling edge/long operation): 1 byte value [0...255]

Sent value [0...255]: 0

Connected contact type

Options: normally closed
normally open

- *normally closed:* The input is opened on actuation.
- *normally open:* The input is closed on actuation.

Long operation after…

Options: 0.3/0.4/0.5/0.6/0.8 s
1/1.2/1.5 s
2/3/4/5/6/7/8/9/10 s

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

Note
The remaining parameter descriptions can be found in Parameter Distinction between short and long operation – no, page 51.

3.2.3 Parameter window *Enable Inputs b…h*

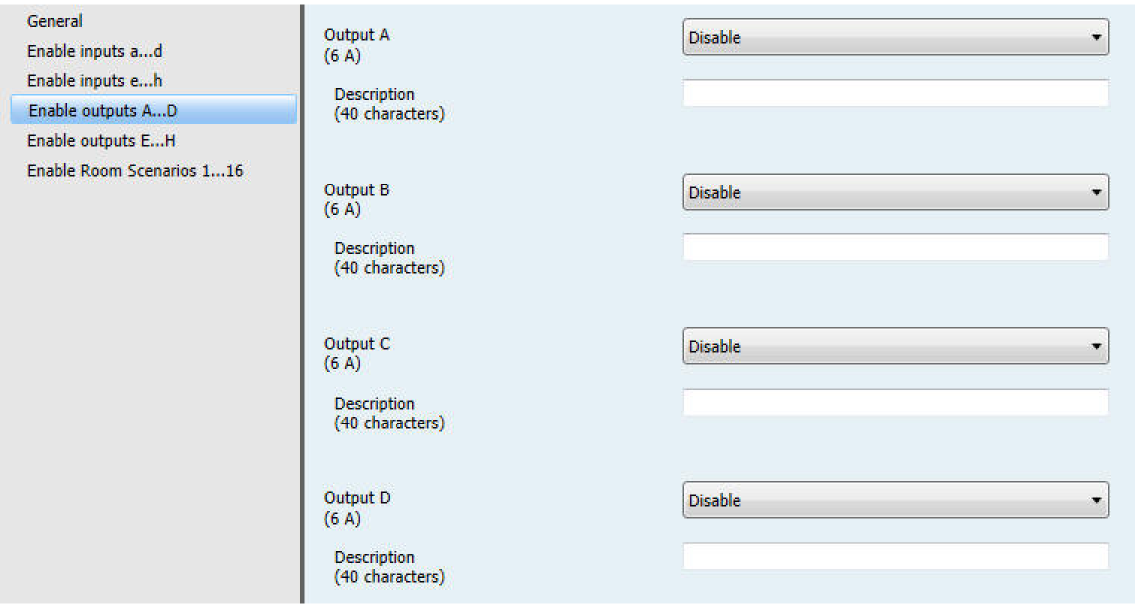
The inputs b...h do not differ from input a.

The descriptions of the parameter setting options and the adjustable communication objects for the inputs g…l can be found in the descriptions at Parameter window Enable Inputs a…h, page 26 and Parameter window a: Switch sensor page 28.

3.2.4 Parameter window *Enable Outputs A…H*

In this parameter window, Outputs A…H are enabled.

Note
In the following, the setting possibilities of Outputs A…H are explained using output A (6 A) as an example. The setting possibilities for outputs A…H are identical.



Output A (6 A): Disable
Description (40 characters):
Output B (6 A): Disable
Description (40 characters):
Output C (6 A): Disable
Description (40 characters):
Output D (6 A): Disable
Description (40 characters):

Output A
(6 A)

Options: Enabled
Disabled

- *Enabled:* The parameter window *A: Output (6 A)* appears. Dependent communication objects become visible.
- *Disabled:* The Output A (6 A) is blocked/invisible, no communication objects are visible.

Description
(40 characters)

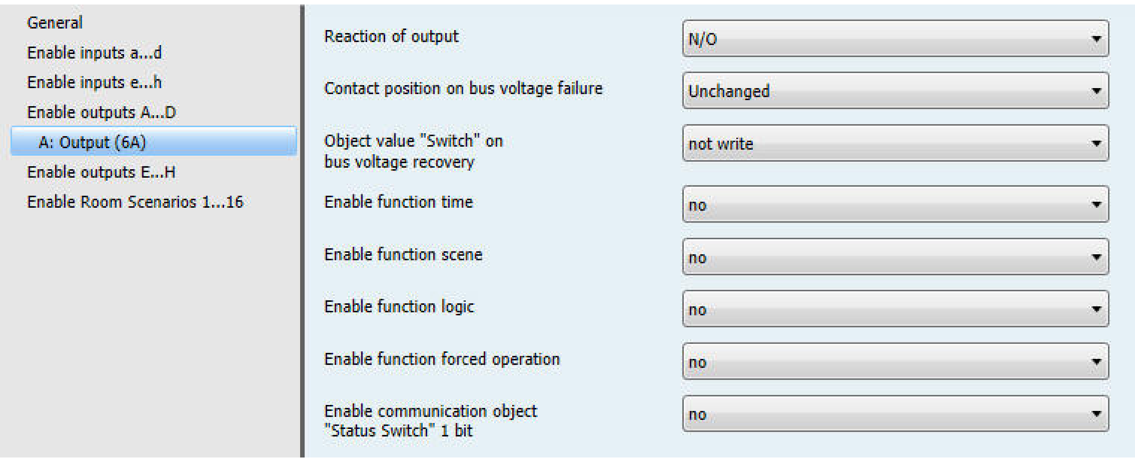
With this parameter, it is possible to enter a text of up to 40 characters in length for identification in the ETS.

Note
The text which is entered is used to provide help, in order to obtain an overview of the inputs when they are fully assigned and to indicate the function assigned to the input. The text is purely for informative purposes and has no further function.

3.2.4.1 Parameter window *A: Output (6 A)*

In this parameter window, all settings for the output A (6 A) are undertaken. The explanations also apply for the Outputs B…H.

This parameter window is visible if in Parameter window Enable Outputs A…H…, page 57, the *Output A (6 A)* has been enabled.



Reaction of output	N/O
Contact position on bus voltage failure	Unchanged
Object value "Switch" on bus voltage recovery	not write
Enable function time	no
Enable function scene	no
Enable function logic	no
Enable function forced operation	no
Enable communication object "Status Switch" 1 bit	no

Reaction of output

Options: N/C
<u>N/O</u>

It can be set in this parameter whether the output operates as a *normally closed contact* or *normally open contact*.

- *N/O:* An ON telegram (1) closes the contact, and an OFF telegram (0) opens the contact.
- *N/C:* An ON telegram (1) opens the contact, and an OFF telegram (0) closes the contact.

Contact position on bus voltage failure

Options: normally closed
normally open
<u>Unchanged</u>

The output can adopt a defined state on bus voltage failure (BVF) using this parameter.

- *normally closed:* The contact is opened with bus voltage failure.
- *normally open:* The contact is closed with bus voltage failure.
- *unchanged:* No change of the contact position.

Note
The reaction on bus voltage failure, recovery and download is to be monitored.

Object value "Switch" on bus voltage recovery

Options: not write
write with 0
write with 1

With this parameter, the output can be influenced by the value of the communication object *Switch* on bus voltage recovery.

The communication object *Switch* can be written with either a 0 or 1 when the bus voltage recovers. The contact position is redefined and set in dependence on the set device parameterization.

- *not write:* The communication object assumes the value 0. This value remains as it is until modified via the bus. The contact position is only re-evaluated and set at this time.

Note
The reaction on bus voltage failure, recovery and download is to be monitored. The Room Master draws the energy for switching the contact from the bus. After bus voltage is applied, sufficient energy is only available after about ten seconds in order to switch all contacts simultaneously. Depending on the set transmission and switching delay after recovery of bus voltage set in the parameter window *General*, the individual outputs will only assume the desired contact position after this time. If a shorter time is set, the RM/S will only switch the first contact when sufficient energy is stored in the Room Master, in order to ensure that enough energy is available to immediately bring all outputs safely to the required position with a renewed bus voltage failure.

Enable function time

Options: no
yes

- *no:* The parameter window remains disabled and invisible.
- *yes:* The parameter window - *Time* appears.

After the function *Time* has been enabled, the parameter window - *Time* is enabled. Further settings can be made here, e.g. on and off delays with staircase lighting.

Note
A more exact description of the function can be found at Communication objects Output A (6 A), page 98, No. 136.

Enable function scene

Options: no
yes

- *no:* The parameter window remains disabled and invisible.
- *yes:* The parameter window - *Scene* appears.

After the function *Scene* has been enabled, the parameter window - *Scene* is enabled. Here you can undertake further settings, e.g. allocation of the output to a scene or standard value.

Enable function logic

Options: no
yes

- *no:* The parameter window remains disabled and invisible.
- *yes:* The parameter window - *Logic* appears.

After the function *Connection/Logic* has been enabled, the parameter window - *Logic* is enabled. Here further settings can be undertaken, e.g. connection and linking of the connection.

Enable function forced operation

Options: no
yes

This parameter enables the function *Forced operation*.

A communication object forced operation is available for every output.

The forced operation (a 1 bit or 2 bit communication object per output) sets the output in a defined state, where – as long as the forced operation is active – it can only be changed via the communication object forced operation.

The switch state after the end of forced operation can be set using the parameter *Contact position with end of the forced operation*.

- *yes:* The following parameters appear:

  Type of object "Forced operation"

  Options: 1 bit
  2 bit

  Using the 2 bit communication object, the output state is defined directly via the communication object value. The control of the output via the communication object *Switch* is blocked as long as the output is forcibly switched ON or OFF.

  The following parameters appear when *1 bit* is selected:

Contact position if forced operation

Options: ON
<u>OFF</u>
Unchanged

- *ON:* Contact position of the output during forced operation.
- *OFF:* Contact position of the output during forced operation.
- *Unchanged:* Contact position of the output during forced operation.

The options *unchanged*, *ON* and *OFF* related to the 1 bit forced operation object and determine the switching state of the output during forced operation. The forced operation relates to the 1 bit forced operation communication object of output X that is available to every output.

Contact position with end of the forced operation

Options: ON
OFF
Unchanged
<u>calculate present contact position</u>

This parameter determines the contact position of the relay after the end of forced operation.

- *ON:* The output is switched ON after forced operation has ended.
- *OFF:* The output is switched OFF after forced operation has ended
- *unchanged:* The contact position is retained during forced operation or safety priority. The contact position only changes when a new calculated switch value is received.
- *calculate present contact position:* After forced operation has ended, the value of the switch position is recalculated and immediately initiated, i.e., the output continues to operate normally in the background during forced operation.

The following parameters appear when *2 bit* is selected:

Contact position with end of the forced operation

Options: ON
OFF
unchanged
calculate present contact position

This parameter determines the contact position of the relay after the end of forced operation.

- *ON:* The output is switched ON after forced operation has ended.
- *OFF:* The output is switched OFF after forced operation has ended
- *unchanged:* The contact position is retained during forced operation or safety priority. The contact position only changes when a new calculated switch value is received.
- *calculate present contact position:* After forced operation has ended, the value of the switch position is recalculated and immediately initiated, i.e., the output continues to operate normally in the background during forced operation.

The telegram value which is sent via the 2 bit communication object determines the switch position as follows:

Value	Bit 1	Bit 0	State	Description
0	0	0	Free	If the communication object *Forced operation* receives a telegram with the value 0 (binary 00) or 1 (binary 01), the output is enabled and can be actuated via different communication objects.
1	0	1	Free	
2	1	0	Forced OFF	If the communication object *Forced operation* receives a telegram with the value 2 (binary 10), the output of the Room Master is forced OFF and remains disabled until forced operation is again deactivated. Actuation via another communication object is not possible as long as the forced operation is activated. The state of the output at the end of forced operation can be programmed.
3	1	1	Forced ON	If the communication object *Forced operation* receives a telegram with the value 3 (binary 11), the output of the Room Master is forced ON and remains disabled until forced operation is again deactivated. Actuation via another communication object is not possible as long as the forced operation is activated.

Enable communication object "Status Switch" 1 bit

Options: no
<u>yes</u>

Caution

If an internal connection with an output is selected, and at the same time the reaction to an event is parameterized with TOGGLE, the communication object *Switch 1* of the binary input is updated with the inverted value of the communication object *Status Switch* of the output.

Ensure that the communication object *Status Switch* of the output is enabled. The settings *normally closed contact/normally open contact* and *Invert status* should be parameterized, so that a TOGGLE function is possible.

- *yes:* The following parameters appear:

Send object value

Options: no, update only
after a change
on request
<u>after a change or on request</u>

- *no, update only:* The status is updated but not sent.
- *after a change:* The status is sent after a change.
- *after request:* The status is sent after a request.
- *after a change or request:* The status is sent after a change or a request.

Object value of contact position

Options: <u>1= closed, 0=open</u>
0 = closed, 1 = opened

With this parameter, the communication object value of the switch status (*Status switch*) is defined.

- *1 = closed, 0 = open:* A closed contact is represented by communication object value 1 and an open contact is represented by the value 0.
- *0 = closed, 1 = open:* A closed contact is represented by communication object value 0 and an open contact is represented by the value 1.

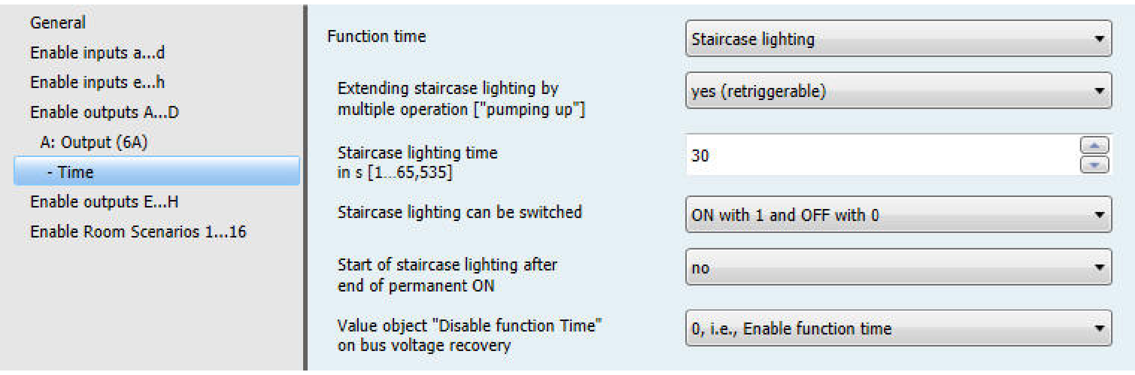
Note

The contact position and thus the switch status can result from a series of priorities and links.

3.2.4.1.1 Parameter window *A: Output (6 A) - Time*

In this parameter window, all settings for the function *Time* are undertaken: Staircase lighting, *Switching ON and OFF delay* and *Flashing*.

This parameter is visible if in parameter window Parameter window A: Output (6 A), page 58, the parameter *Enable function time* has been enabled.



Function time: Staircase lighting

Extending staircase lighting by multiple operation ["pumping up"]: yes (retriggerable)

Staircase lighting time in s [1...65,535]: 30

Staircase lighting can be switched: ON with 1 and OFF with 0

Start of staircase lighting after end of permanent ON: no

Value object "Disable function Time" on bus voltage recovery: 0, i.e., Enable function time

Explanations concerning the time functions and the timing sequences can be found in Planning and application, page 101. Please observe the Function diagram, page 102, from which the switching and timing priorities originate.

Note
Observe the contact life span and switching cycles per minute. For the contact life and switching operations per minute, see Technical data, page 9.

Function *Time*

Options: Staircase lighting
Delay for switching ON and OFF
Flashing

This parameter defines the type of function *Time* for each output.

- *Staircase lighting:* The value, with which the staircase lighting is switched on and off, can be parameterized. The staircase lighting time is started when the function is activated. It is switched off immediately after the staircase lighting time has been completed.

Note
The function *Staircase lighting* can also be called via the communication object *Switch, Logical connection x* (x = 1, 2) or called with a light scene recall.

- *Delay for switching ON and OFF*: The output can be switched on or off with a delay via this function.
- *Flashing:* The output starts to flash as soon as the parameterized value is received in the communication object *Switch*. The flashing period can be adjusted via the parameterized time duration for ON or OFF. The output is switched on at the start of the flashing period. At the start of the flashing period, the output is switched on with a normally open contact and off with a normally closed contact. When a new value is received on the communication object *Switch*, the flashing period will recommence. The relay state after flashing can be programmed. Flashing can be inverted when the output is used as a normally closed contact. The communication object *Status switch* indicates the current relay state during flashing.

The following parameter appears with the selection *Staircase lighting*:

**Staircase lighting time
in s [1…65,535]**

Options: 1…30…65,535

The staircase lighting defines how long the contact is closed – provided that the contact is programmed as a normally open contact – and how long the light remains on after an ON telegram. The input is made in seconds.

**Extending staircase lighting by
multiple operation ["pumping up"]**

Options: no (not retriggerable)
yes (retriggerable)
up to max. 2 x staircase lighting time
up to max. 3 x staircase lighting time
up to max. 4 x staircase lighting time
up to max. 5 x staircase lighting time

If a further ON telegram is received during the staircase lighting time sequence, the remaining staircase lighting time can be extended by a further period. This is possible by repeated operation of the button (“Pumping up“) until the maximum programmed number of retriggering operations is reached. The maximum time can be set to 1, 2, 3, 4 or 5-fold time of the staircase lighting time.

The staircase lighting time is extended by “Pumping up” to the maximum time. If some of the time has already timed out, the staircase lighting time can again be extended to the maximum time by “pumping up”. The parameterized maximum time may not however be exceeded.

- *no:* The receipt of an ON telegram is ignored. The staircase lighting time continues without modification to completion.
- *yes (retriggerable):* The staircase lighting time is reset each time by a renewed ON telegram and starts to count again. This process can be repeated as often as desired using this selection.
- *up to max. 2/3/4/5 x staircase lighting time:* The staircase lighting time is extended by the 2/3/4/5-fold staircase lighting time with renewed ON telegrams.

Staircase lighting can be switched

Options: ON with 1 and OFF with 0
ON with 1 no action with 0
ON with 0 or 1, switch OFF not possible

This parameter defines the telegram value used for switching the staircase lighting on and off prematurely.

- *ON with 0 or 1, switch OFF not possible:* The function *Staircase lighting* is switched on independently of the value of the incoming telegram. Premature switch off is not possible.

Restart of staircase lighting after end of permanent ON

Options: no
yes

- *no:* The lighting switches off if *Permanent ON* is ended.
- *yes:* The lighting remains on and the staircase lighting time restarts.

The function of continuously ON is controlled via the communication object *Permanent ON*. If the communication object receives a telegram with the value 1, the output is switched ON regardless of the value of the communication object *Switch* and remains switched on until the communication object *Permanent ON* has the value 0.

Value object "Disable function Time" on bus voltage recovery

Options: unchanged
1, i.e., Disable function time
0, i.e., Enable function time

This parameter defines how the parameter function *Time* should behave after bus voltage recovery. With a telegram to the communication object *Disable function time*, the function *Time* can be disabled.

- *unchanged:* The function *Time* can continue unchanged.

Note
The state *Function time* is stored with bus voltage failure and continues unchanged after bus voltage recovery.

- *1 = disable function time:* The function *Time* is disabled by a telegram with the value 1.

Note
They can only be enabled via the communication object *Disable function time*.

- *0 = enable function time:* The function *Time* is enabled by a telegram with the value 0.

Note
If the staircase lighting is disabled when the function *Time* is operational, the light will stay at ON until it is switched to OFF manually.

How does the staircase lighting behave with bus voltage failure?

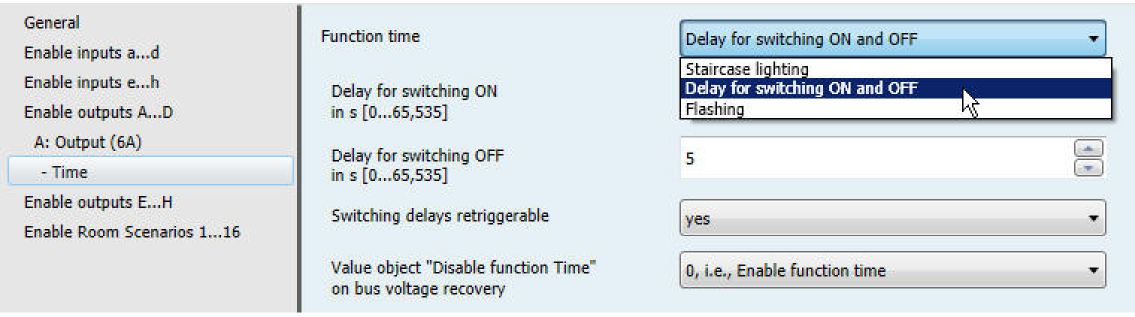
The behaviour at bus voltage failure is determined by the parameter *Reaction on bus voltage failure* in the parameter window *A: Output (6 A).*

How does the staircase lighting behave with bus voltage recovery?

The behaviour at bus voltage recovery is defined by two conditions.

1. By the communication object *Disable function time*. If the staircase lighting is blocked after bus voltage recovery, the staircase lighting can only be switched on or off via the communication object *Switch*.
2. By the parameterization of the communication object *Switch*. Whether the light is switched on or off with bus voltage recovery depends on the programming of the communication object *Switch*.

The following parameters appear with *Delay for switching ON and OFF*:



Function time: Delay for switching ON and OFF

Delay for switching ON in s [0...65,535]

Delay for switching OFF in s [0...65,535]: 5

Switching delays retriggerable: yes

Value object "Disable function Time" on bus voltage recovery: 0, i.e., Enable function time

Explanations for the ON and OFF delay can be found at Delay for switching ON and OFF, page 105. You will also find a timing diagram as well as explanations on the effect of various ON and OFF telegrams in combination with the switching ON and OFF delay.

Delay for switching ON
in s [0...65,535]

Options: 0…5…65,535

Here you set the time by which an ON telegram is delayed after switch on.

Delay for switching OFF
in s [0...65,535]

Options: 0…5…65,535

Here you set the time by which switch OFF is delayed after a switch OFF telegram.

Switching delays retriggerable

Options: no
yes

- *no:* The delay time cannot be retriggered.
- *yes:* The delay time can be retriggered.

Value object "Disable function Time"
on bus voltage recovery

Options: unchanged
1, i.e., Disable function time
0, i.e., Enable function time

This parameter defines how the function *Time* should behave after bus voltage recovery. With a telegram to the communication object *Disable function time,* the function *Time* can be disabled or enabled.

- *unchanged:* After bus voltage recovery, the function *Time* reacts in the same way as before bus voltage failure.
- *1 = disable function time:* The function *Time* is disabled by a telegram with the value 1.
- *0 = enable function time:* The function *Time* is enabled by a telegram with the value 0.

How does the staircase lighting behave with bus voltage failure?

The behaviour at bus voltage failure is determined by the parameter *Reaction on bus voltage failure* in the parameter window *A: Output (6 A).*

How does the staircase lighting behave with bus voltage recovery?

The behaviour at bus voltage recovery is defined by two conditions.

1. By the communication object *Disable function time*. If the staircase lighting is blocked after bus voltage recovery, the staircase lighting can only be switched on or off via the communication object *Switch*.
2. By the parameterization of the communication object *Switch*. Whether the light is switched on or off with bus voltage recovery depends on the programming of the communication object *Switch*.

The following parameter appears with the selection *Flashing*:

Flashing if communication object "Switch" is

Options: ON (1)
OFF (0)
ON (1) or OFF (0)

Here you set the value of the communication object *Switch* at which the output flashes. Flashing is not retriggerable.

- *ON (1):* Flashing starts when a telegram with the value 1 is received on the communication object *Switch*. A telegram with the value 0 ends flashing.
- *OFF (0):* Flashing starts when a telegram with the value 0 is received on the communication object *Switch*. A telegram with the value 1 ends flashing.
- *ON (1) or OFF (0):* A telegram with the value 1 or 0 triggers flashing. Suspension of flashing is not possible in this case.

Duration for ON in value x 0.1 s [5...65,535]

Options: 5…10…65,535

This parameter defines how long the output is switched ON during a flashing period.

Duration for OFF in value x 0.1 s [5...65,535]

Options: 5…10…65,535

This parameter defines how long the output is switched off during a flashing period.

Number of impulses [1...100]

Options: 1…5…100

This parameter defines the maximum number of pulses. This is useful to avoid unnecessary wear of the contacts caused by flashing.

Contact position after flashing

Options: ON
OFF
<u>calculate present contact position</u>

This parameter defines the state that the parameter should assume after flashing.

- *ON:* The output is switched on after flashing.
- *OFF:* The output is switched off after flashing.
- *calculate present contact position:* The output assumes the switching state which it had before flashing commenced.

For further information see: Function diagram, page 102.

Value object "Disable function Time" on bus voltage recovery

Options: unchanged
<u>1, i.e., Disable function time</u>
0, i.e., Enable function time

This parameter defines how the *Function time* should behave after bus voltage recovery. With a telegram to the communication object *Disable function time*, the function *Time* can be disabled.

- *unchanged:* After bus voltage recovery, the function *Time* reacts in the same way as before bus voltage failure.
- *1 = disable function time:* The function *Time* is disabled by a telegram with the value 1.
- *0 = enable function time:* The function *Time* is enabled by a telegram with the value 0.

How does the staircase lighting behave with bus voltage failure?

The behaviour at bus voltage failure is determined by the parameter *Reaction on bus voltage failure* in the parameter window *A: Output (6 A).*

How does the staircase lighting behave with bus voltage recovery?

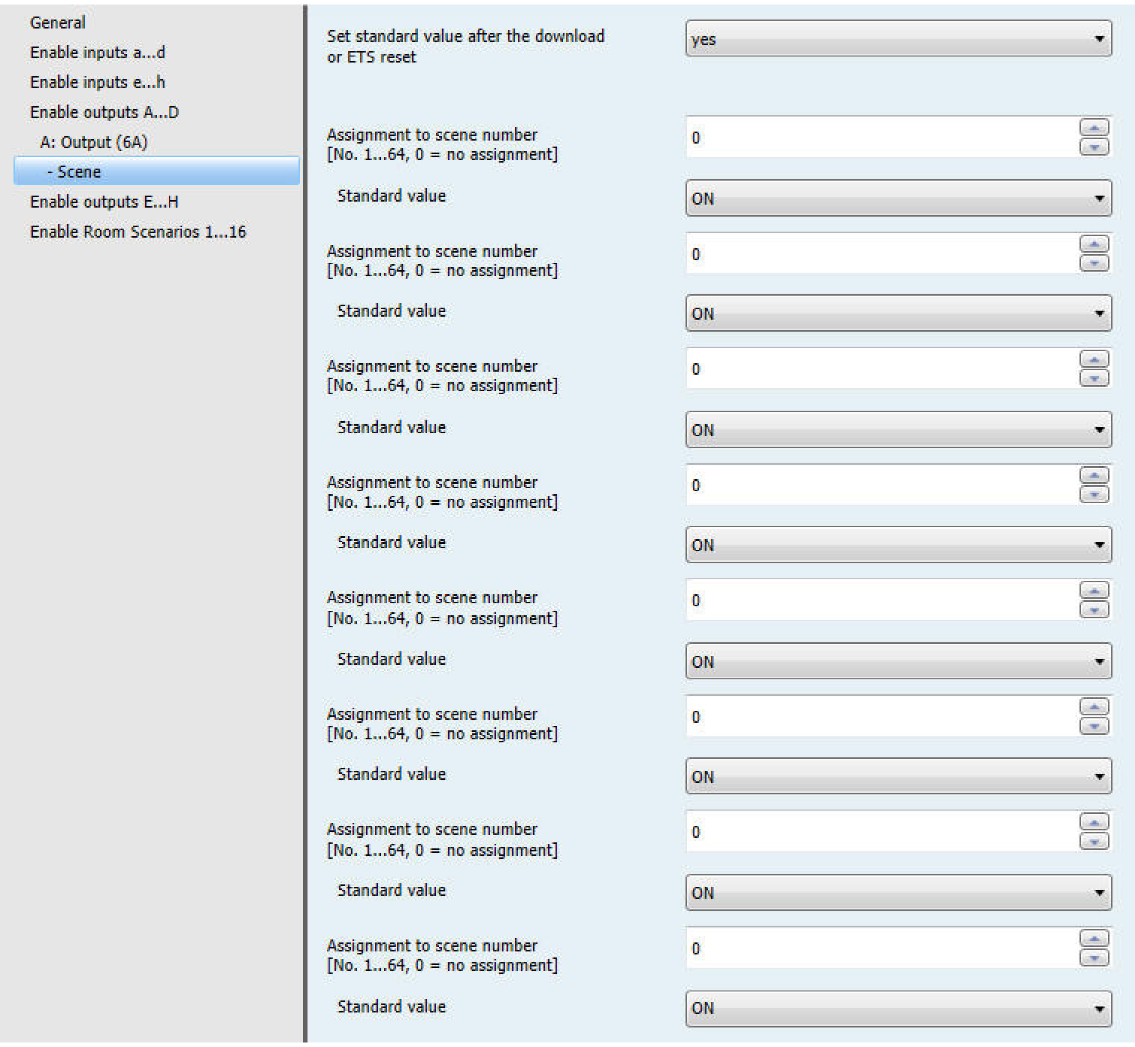
The behaviour at bus voltage recovery is defined by two conditions.

3. By the communication object *Disable function time*. If the staircase lighting is blocked after bus voltage recovery, the staircase lighting can only be switched on or off via the communication object *Switch*.
4. By the parameterization of the communication object *Switch*. Whether the light is switched on or off with bus voltage recovery depends on the programming of the communication object *Switch*.

3.2.4.1.2 Parameter window *A: Output (6 A) - Scene*

In this parameter window, all settings for the function *Scene* are undertaken.

This parameter window is visible if in Parameter window A: Output (6 A), page 58, the parameter *Enable function time* has been enabled.



Set standard value after the download or ETS reset	yes
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON
Assignment to scene number [No. 1...64, 0 = no assignment]	0
Standard value	ON

How is a scene set?

Via the communication object *Scene*

- Sets the value for standard values.
- The scene can be recalled.
- The scene can be changed.
- The scene can be saved.

An example:

Scene recall:

- Send value 0…63 for the scene (no. 1…64) to the communication object *Scene*.

Scene change and save:

- Scene no. 24 is assigned to the output with the value ON.
- Scene no. 24 should be assigned to the output with the value OFF:
    - Set the output to OFF with a switch telegram.
    - Send value 151 (128 + 23) for storage of scene number 24 to the communication object *Scene*.

General values for scene storage:

- 128 + (0…63) for the scene (No. 1…64)
    - The stored scene values are retained until there is a device reset.

Note
After a device reset, the parameterized values can be reactivated. **For further information see: ETS reset, page 111.**

Set standard value after the download or ETS reset

Options: no
yes

- *no:* The standard values are not set after a download or ETS reset.
- *yes:* The standard values are set after a download or ETS reset.

Assignment to scene number [No. 1...64, 0 = no assignment]

Options: 0…64

Using the function *Scene*, up to 64 scenes are managed using just a single group address. With this group address, all slaves integrated into a scene are linked via a 1 byte communication object. The following information is contained in a telegram:

- Number of the scene (1…64) as well as
- Telegram: Call scene or store scene.

The output can be integrated in up to eight scenes. So for example, the scene can be switched on in the morning and switched off in the evening, or the output can be integrated into light scenes.

If a telegram is received on the communication object *Scene*, the sent scene number is allocated for all outputs, which carry out the stored scene position, or the current position is stored as the new scene position.

Standard value

Options: ON
OFF

Here you set the state that the output has when the scene is recalled.

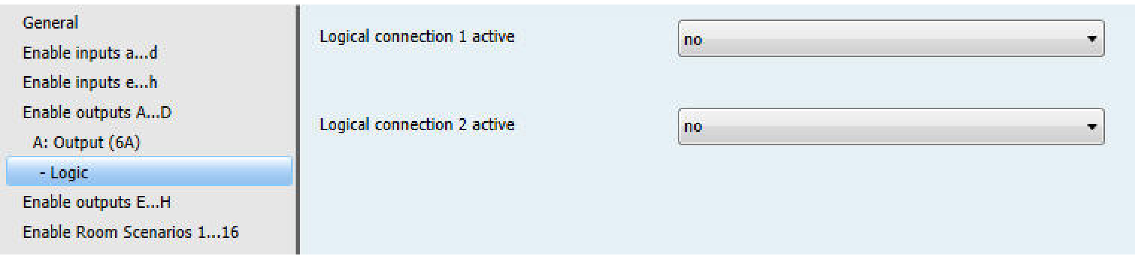
Note
When a scene is recalled: • the function *Time* is restarted. • the logical connections are re-evaluated.

For further information see: Communication objects Output A (6 A), page 98, Function Scene, page 109 and Code table scene (8 bit), page 124.

3.2.4.1.3 Parameter window *A: Output (6 A) - Logic*

In this parameter window, all settings for the function *Function Logic* are undertaken.

This parameter window is visible if in Parameter window A: Output (6 A), page 58, the parameter *Enable function time* has been enabled.



Logical connection 1 active: no

Logical connection 2 active: no

The function *Function Logic* provides up to two logic objects for each output, which can be logically linked with the communication object *Switch*.

The logic is always re-calculated when a communication object value is received. Hereby, the communication object *Logical connection 1* is first of all evaluated with the communication object *Switch.* The result is then logically linked with the communication object *Logical connection 2*.

Explanation for the logical function can be found at Connection/logic, page 107. Please also observe the Function diagram, page 102, where the priorities become evident.

Logical connection 1 active

Options: no
yes

With these parameters, the communication object *Logical connection 1* is enabled.

- *yes:* The following parameters appear:

Function of logical connection

Options: AND
OR
XOR
GATE

The logical function of the communication object *Logical connection 1* is determined with the switch telegram. All three standard operations (AND, OR, XOR) are possible. Furthermore, the GATE operation can be used to inhibit switch commands.

For further information see: Connection/logic, page 107.

Result is inverted

Options: no
yes

- *yes:* The result of the logical connection can be inverted.
- *no:* There is no inversion.

**Object value "Logical connection 1"
after bus voltage recovery**

Options: not write
write with 0
write with 1

This parameter defines the value allocated to the communication object *Logical connection 1* with bus voltage recovery.

- *not write:* after bus voltage recovery, the value 0 remains in the communication object *Switch*. This value remains as it is until the communication object is modified via the bus. The contact position is only re-evaluated and set at this time. The correct status of the contact position is displayed via the communication object *Status switch* independently of the value of the communication object *Switch*.

Note
The values of the communication objects *Logical connection 1/2* are stored at bus voltage failure. The values are set again after a bus voltage recovery If values are not assigned for communication objects *Logical connection 1/2*, they will be deactivated. At a reset via the bus, the values of the communication objects *Logical connection 1/2* remain unchanged.

A further parameter appears if GATE is selected with the parameter *Function of logical connection*:

**Gate disabled, if object value
"Logical connection 1" is**

Options: 1
0

This parameter defines the value, at which the communication object *Logical connection 1* disables the GATE.

Disabling of the gate means that the telegrams received on the communication object *Switch* are ignored. As long as the GATE is activated, the value which was sent last to the input of the GATE remains on the output. After a GATE is blocked, the value that was on the output before the block remains on the output of the GATE.

After the GATE is enabled, this value will be retained until a new value is received.

For further information see: Function diagram, page 102.

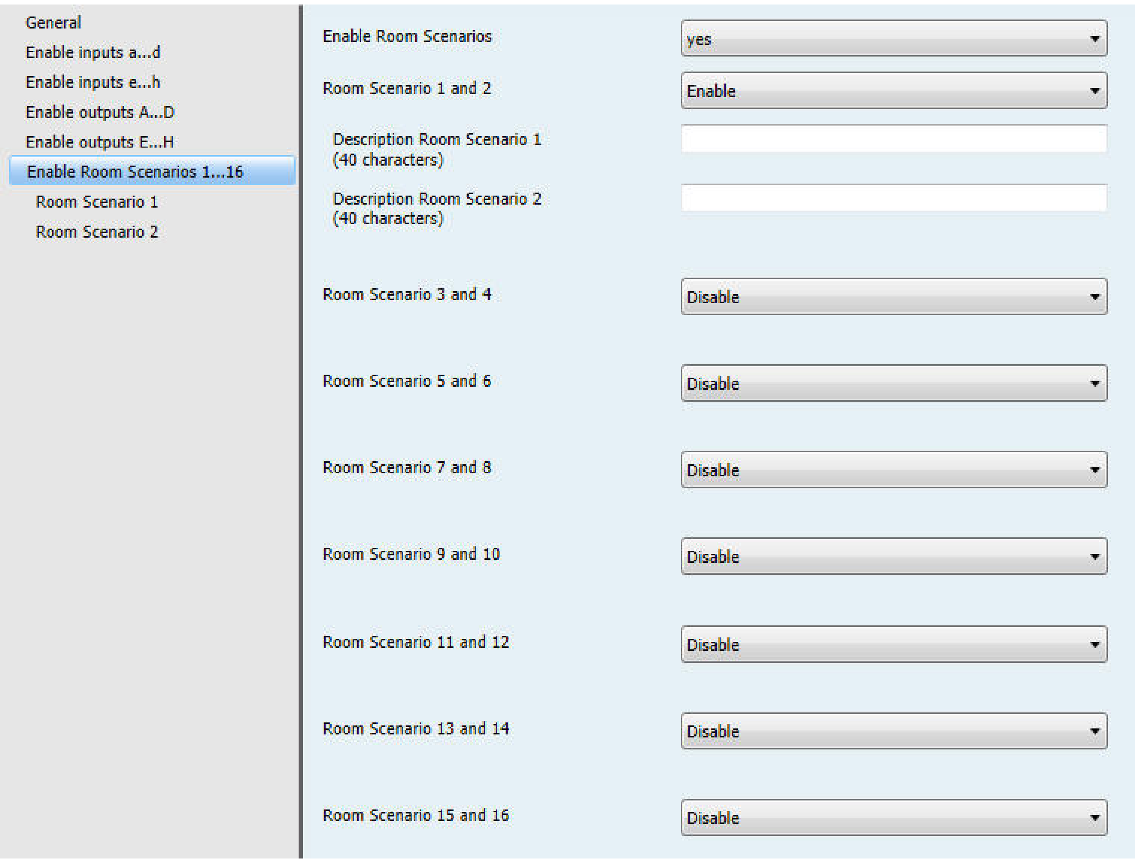
The GATE is disabled after bus voltage failure and remains deactivated after bus voltage recovery.

Logical connection 2 active

The same programming options exist as those for parameter *Logical connection 1 active*.

3.2.5 Parameter window *Enable Room Scenarios 1…16*

In this parameter window, the Room Scenarios 1…16 can be enabled in pairs and assigned with a description.



Enable Room Scenarios	yes
Room Scenario 1 and 2	Enable
Description Room Scenario 1 (40 characters)	
Description Room Scenario 2 (40 characters)	
Room Scenario 3 and 4	Disable
Room Scenario 5 and 6	Disable
Room Scenario 7 and 8	Disable
Room Scenario 9 and 10	Disable
Room Scenario 11 and 12	Disable
Room Scenario 13 and 14	Disable
Room Scenario 15 and 16	Disable

Enable Room Scenarios

Options:	no
	yes

With this parameter, the Room Scenarios 1…16 as well as the seven communication objects No. 2…8 are enabled.

Note
In the following parameters, the Room Scenarios 1…16 are represented by x and y, as the functions for all Room Scenarios are the same. Here x represents the oddly numbered room scenarios 1/3/5/7/9/11/13 or 15, and y represents the evenly numbered room scenarios 2/4/6/8/10/12/14 or 16.

Room Scenario x and y

Options: Enable
<u>Disabled</u>

- *Disabled:* The Room Scenarios x/y are disabled.
- *Enable:* The Room Scenarios x/y are enabled. They are triggered by the receipt of a telegram on the communication object no. 2. The parameter windows *Room Scenario x* and *Room Scenario y* also appear. The following parameters also appear:

Description Room Scenario x (40 characters)

Options: - - - TEXT - - -

With this parameter, it is possible to enter a text of up to 40 characters in length for identification or the Room Scenario in the ETS.

Description Room Scenario y (40 characters)

Options: - - - TEXT - - -

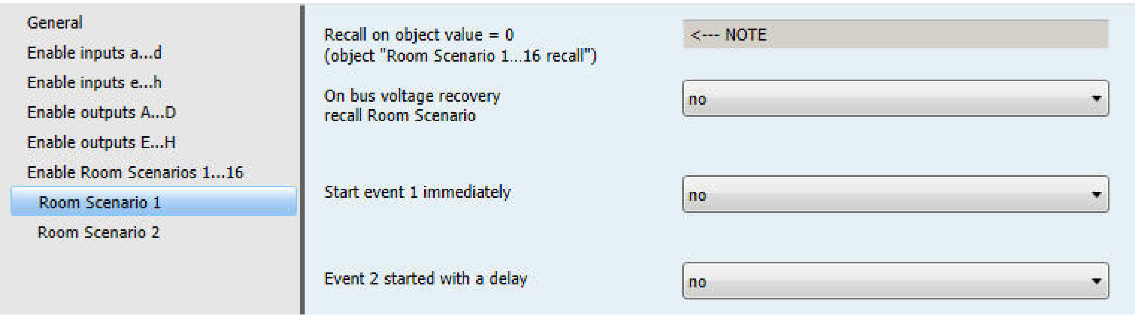
With this parameter, it is possible to enter a text of up to 40 characters in length for identification or the Room Scenario in the ETS.

Note
The entered text is used as to assist in providing an overview of the Room Scenarios and the functions they involve. It has no other function.

3.2.5.1 Parameter window *Room Scenario x*

This parameter window is visible if in parameter window *Enable Room Scenarios 1…16*, the option *yes* is selected with *Enable Room Scenarios* as well as with parameter *Room Scenarios x and y* and the option *Enable* has been selected.

Note
In the following parameters, the Room Scenarios 1…16 are represented by x and y, as the functions for all Room Scenarios are the same. Here x represents the oddly numbered room scenarios 1/3/5/7/9/11/13 or 15, and y represents the evenly numbered room scenarios 2/4/6/8/10/12/14 or 16.



Recall on object value = 0
(object "Room Scenario 1...16 recall") <--- NOTE

On bus voltage recovery
recall Room Scenario no

Start event 1 immediately no

Event 2 started with a delay no

Recall on object value = 0
(object "Room Scenario 1…16 recall")

<--- NOTE

The Room Scenarios are triggered via communication object no. 2 *Room Scenario 1…16 recall*, i.e. *Room Scenario 1* is triggered when a 0 is received. *Room Scenario 2* when a 1 is received, etc.

For further information see: Communication objects General, page 88, and Room Scenario external triggering, page 118

The Room Scenarios can also be internally triggered via binary inputs. It is important to note that the Room Scenarios are always triggered in pairs, e.g. *Room Scenario 5* when a 0 is received and *Room Scenario 6* when a 1 is received.

For further information see: Communication objects General, page 88, and Room Scenario external triggering, page 118

On bus voltage recovery
recall Room Scenario

Options: <u>no</u>
yes

Using this parameter, the reaction after bus voltage recovery is set.

- *no:* After bus voltage recovery, the state, which existed before bus voltage failure, is set.
- *yes:* This Room Scenario is triggered after bus voltage recovery.

Start event 1 immediately

Options: no
yes

- *no:* This is no reaction, when the value 0 is received. Event 1 is not started.
- *yes:* If the value 0 is received, event 1 starts. Event 1 is set via the following parameters:

recall scene

Options: no
only device internal
only via the bus
device internal and via the bus

This parameter defines how and where a scene recall is sent with the start of event 1 via communication object no. 6 *Room Scenario Recall KNX scene*.

- *only device internal:* The set scene number is only recalled internally in the device, e.g. in order to trigger a determined room scenario.
- *only via the bus:* The set scene number is sent via the bus. Accordingly, further KNX devices can be integrated into the Room Scenario, or these are also contacted by a scene recall.
- *device internal and via the bus:* The set scene number is recalled both device internally as well as being sent via the bus. Thus, a Room Scenario can be triggered, and further KNX devices integrated into the scene can be contacted.

Scene number [1…64]

Options: 1…64

This parameter defines the scene number, which is to be triggered by a scene recall. 64 scene numbers are available.

Switch 1 send

Options: no
ON
OFF
TOGGLE

This parameter defines the value at which communication object no. 3 should send a telegram.

- *no:* There is no reaction with the start of the event.
- *ON:* A telegram with the value 1 is sent on the bus via communication object no. 3.
- *OFF:* A telegram with the value 0 is sent on the bus via communication object no. 3.
- *TOGGLE:* Via the communication object no. 3, a telegram is sent with the opposite value, e.g. if the value 1 was read beforehand, when the event 1 is recalled the value 0 is sent, and vice versa.

Switch 2 send

Options: no
ON
OFF
TOGGLE

Options:	<u>no</u>
	ON
	OFF
	TOGGLE

This parameter defines the value at which communication object no. 4 should send a telegram.

- *no:* There is no reaction with the start of the event.
- *ON:* A telegram with the value 1 is sent on the bus via communication object no. 4.
- *OFF:* A telegram with the value 0 is sent on the bus via communication object no. 4.
- *TOGGLE:* Via the communication object no. 4, a telegram is sent with the opposite value, e.g. if the value 1 was read beforehand, when the event 1 is recalled the value 0 is sent, and vice versa.

ON/OFF send to thermostat

Options:	<u>no</u>
	ON
	OFF

This parameter defines whether a thermostat, e.g. RDF/A is switched on or off, or whether it remains in an unchanged state.

- *no:* There is no reaction with the start of the event.
- *ON:* A telegram with the value 1 is sent on the bus via communication object no. 8.
- *OFF:* A telegram with the value 0 is sent on the bus via communication object no. 8.

Send 1 byte value

Options:	<u>no</u>
	Value [0…255]

This parameter defines whether a 1 byte value is sent.

- *Value [0…255]:* The following parameter appears:

Sent value

Options: 0…255

A telegram with the respective value is sent on the bus via communication object no. 9.

Automatic blind triggering

Options: <u>no</u>
yes

- *no:* There is no reaction with the start of the event.
- *yes:* The telegram for automatic activation is sent on the bus. The KNX devices integrated into the automatic function are also contacted.

Internal blocking of the inputs

Options: <u>unchanged</u>
Activate
Deactivate

This parameter acts directly on the binary inputs, which allow an internal block.

- *unchanged:* The internal block remains unchanged.
- *Activate:* The internal block is activated.
- *Deactivate:* The internal block is deactivated.

Event 2 started with a delay

Options: <u>no</u>
yes

- *no:* This is no reaction, when the value 0 is received. Event 2 is not started.
- *yes:* If the value 0 is received, event 2 starts. Event 2 is set via the following parameters:

**Delay time
in s [0…65,535]**

Options: 0…<u>30</u>…65,535

This parameter determines the duration, after which event 2 is started.

Note
The following parameters and their descriptions do not differ from those with the description Start event 1 immediately, page 81.

3.2.6 Commissioning without bus voltage

How is the device switched on and put into operation?

The device can be made operational by applying an auxiliary voltage from the mobile power supply (NTI).

3.3 Communication objects

Note

As standard, the write flag (with the exception of 1 bit communication objects) is deleted with the communication object values. Thus the communication object value cannot be changed via the bus. If this function is required, the Write flag must be set in the ETS.

The communication object value is overwritten with the parameterized value after bus voltage recovery.

3.3.1 Short overview of the communication objects

CO No.	Function	Name	Data Point Type (DPT)	Length	Flags				
					C	R	S	T	R
0	In operation	System	1.002	1 bit	x			x	
1	Request status values	General	1.017	1 bit	x		x		
2	Recall 1...16	Room Scenario	17.001	1 byte	x		x		
3	Switch 1	Room Scenario	1.001	1 bit	x		x	x	
4	Switch 2	Room Scenario	1.001	1 bit	x		x	x	
5	Automatic blind triggering	Room Scenario	1.001	1 bit	x			x	
6	Recall KNX scene	Room Scenario	18.001	1 byte	x			x	
7	Trigger internal block	Room Scenario	1.001	1 bit	x			x	
8	Thermostat ON/OFF	Room Scenario	1.001	1 bit	x			x	
9	Send value	Room Scenario	5.010	1 byte	x			x	

CO No.	Function	Name	Data Point Type (DPT)	Length	Flags				
					C	R	S	T	R
10	Block	Input a: Switch sensor	1.003	1 bit	x		x		
		Input a: Switch/dimming sensor	1.003	1 bit	x		x		
		Input a: Blind sensor	1.003	1 bit	x		x		
		Input a: Value/forced operation	1.003	1 bit	x		x		
11	Switch 1	Input a: Switch sensor	1.001	1 bit	x		x	x	
	Switch	Input a: Switch/dimming sensor	1.001	1 bit	x		x	x	
	Blind UP/DOWN	Input a: Blind sensor	1.008	1 bit	x		x	x	
	Value 1, unsigned	Input a: Value/forced operation	8.001	2 byte	x			x	
	Value 1, floating point	Input a: Value/forced operation	9.001	2 byte	x			x	
	Value 1, forced operation	Input a: Value/forced operation	2.001	2 bit	x			x	
	Value 1, signed	Input a: Value/forced operation	13.001	4 byte	x			x	
	Value 1, signed	Input a: Value/forced operation	6.010	1 byte	x			x	
	Value 1, unsigned	Input a: Value/forced operation	5.010	1 byte	x			x	
	Value 1, scene number	Input a: Value/forced operation	18.001	1 byte	x			x	
	Value 1	Input a: Value/forced operation	1.001	1 bit	x			x	
	Value 1, time, weekday	Input a: Value/forced operation	10.001	3 byte	x			x	
	Value 1, signed	Input a: Value/forced operation	7.001	2 byte	x			x	
	Value 1, unsigned	Input a: Value/forced operation	12.001	4 byte	x		x	x	
12	Switch 2	Input a: Switch sensor	1.001	1 bit	x		x	x	
	Dimming	Input a: Switch/dimming sensor	3.007	4 bit	x			x	
	STOP/slat adjustment	Input a: Blind sensor	1.007	1 bit	x			x	
	Value 2, unsigned	Input a: Value/forced operation	8.001	2 byte	x			x	
	Value 2, floating point	Input a: Value/forced operation	9.001	2 byte	x			x	
	Value 2, forced operation	Input a: Value/forced operation	2.001	2 bit	x			x	
	Value 2, signed	Input a: Value/forced operation	13.001	4 byte	x			x	
	Value 2, signed	Input a: Value/forced operation	6.010	1 byte	x			x	
	Value 2, unsigned	Input a: Value/forced operation	5.010	1 byte	x			x	
	Value 2, scene number	Input a: Value/forced operation	18.001	1 byte	x			x	
	Value 2	Input a: Value/forced operation	1.001	1 bit	x			x	
	Value 2, time, weekday	Input a: Value/forced operation	10.001	3 byte	x			x	
	Value 2, signed	Input a: Value/forced operation	7.001	2 byte	x			x	
	Value 2, unsigned	Input a: Value/forced operation	12.001	4 byte	x			x	
13	Switch 3	Input a: Switch sensor	1.001	1 bit	x		x	x	
	Upper limit position	Input a: Blind sensor	1.002	1 bit	x		x		
14	Start event 0/1	Input a: Switch sensor	1.001	1 bit	x		x		
	Lower limit position	Input a: Blind sensor	1.002	1 bit	x		x		
15…49	the same CO as input a	Input b…h							

CO No.	Function	Name	Data Point Type (DPT)	Length	Flags				
					C	R	S	T	R
50	Switch	Output A (6 A)	1.001	1 bit	x		x		
51	Permanent ON	Output A (6 A)	1.003	1 bit	x		x		
52	Disable function time	Output A (6 A)	1.003	1 bit	x		x		
53	Scene	Output A (6 A)	18.001	1 byte	x		x		
54	Forced operation	Output A (6 A)	2.001	2 bit	x		x		
	Forced operation	Output A (6 A)	1.003	1 bit	x		x		
55	Status switch	Output A (6 A)	1.001	1 bit	x	x		x	
56	Logical connection 1	Output A (6 A)	1.002	1 bit	x		x		
57	Logical connection 2	Output A (6 A)	1.002	1 bit	x		x		
58…113	the same CO as output A	Output B…H (6 A)							
114…	Not assigned								

3.3.2 Communication objects General

<table>
<tr><th>No.</th><th>Function</th><th>Object name</th><th>Data type</th><th>Flags</th></tr>
<tr><td><b>0</b></td><td><b>In operation</b></td><td><b>System</b></td><td><b>1 bit</b>
<b>DPT 1.002</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">The communication object is enabled if in parameter window <i>General</i> the parameter <i>Send communication object "In operation"</i> has been selected with option <i>yes</i>.
In order to regularly monitor the presence of the device on the KNX, an in operation monitoring telegram can be sent cyclically on the bus.
As long as the communication object is activated, it sends a programmable in operation telegram.
Telegram value 1 = system in operation with option <i>send value 1 cyclically</i>
0 = system in operation with option send value 0 cyclically</td></tr>
<tr><td><b>1</b></td><td><b>Request status values</b></td><td><b>General</b></td><td><b>1 bit</b>
<b>DPT 1.017</b></td><td><b>C, W</b></td></tr>
<tr><td colspan="5">The communication object is enabled if in parameter window <i>General</i> the parameter <i>Enable communication object "Request status values" 1 bit</i> has been selected with option <i>yes</i>.
If a telegram with the value x (x = 0; 1; 0 or 1) is received in the communication object, all status objects are sent on the bus, as long as these have not been programmed with the option <i>after a change</i> or <i>after request</i> or <i>after a change or request</i>.
The following function results for the option x = 1:
Telegram value: 1 = All status messages are sent.
0 = nothing happens.</td></tr>
</table>

3.3.3 Communication objects *Room Scenario*

<table>
<tr><th>No.</th><th>Function</th><th>Object name</th><th>Data type</th><th>Flags</th></tr>
<tr><td><b>2</b></td><td><b>Recall 1...16</b></td><td><b>Room Scenario</b></td><td><b>1 byte</b>
<b>DPT 17.001</b></td><td><b>C, W</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>Enable Room Scenarios 1…16,</i> the parameter Enable <i>Room Scenarios</i> has been selected with the option <i>yes</i>.
<table>
<tr><th colspan="2">1 byte value [0…255]</th><th>DPT 5.010 value</th></tr>
<tr><td>Value 0 =</td><td>Room Scenario 1</td><td>00000000</td></tr>
<tr><td>Value 1 =</td><td>Room Scenario 2</td><td>00000001</td></tr>
<tr><td>Value 2 =</td><td>Room Scenario 3</td><td>00000010</td></tr>
<tr><td>Value 3 =</td><td>Room Scenario 4</td><td>00000011</td></tr>
<tr><td>Value 4 =</td><td>Room Scenario 5</td><td>00000100</td></tr>
<tr><td>Value 5 =</td><td>Room Scenario 6</td><td>00000101</td></tr>
<tr><td>Value 6 =</td><td>Room Scenario 7</td><td>00000110</td></tr>
<tr><td>Value 7 =</td><td>Room Scenario 8</td><td>00000111</td></tr>
<tr><td>Value 8 =</td><td>Room Scenario 9</td><td>00001000</td></tr>
<tr><td>Value 9 =</td><td>Room Scenario 10</td><td>00001001</td></tr>
<tr><td>Value 10 =</td><td>Room Scenario 11</td><td>00001010</td></tr>
<tr><td>Value 11 =</td><td>Room Scenario 12</td><td>00001011</td></tr>
<tr><td>Value 12 =</td><td>Room Scenario 13</td><td>00001100</td></tr>
<tr><td>Value 13 =</td><td>Room Scenario 14</td><td>00001101</td></tr>
<tr><td>Value 14 =</td><td>Room Scenario 15</td><td>00001110</td></tr>
<tr><td>Value 15 =</td><td>Room Scenario 16</td><td>00001111</td></tr>
</table>
Sending a value from 16 to 255 is invalid and will be ignored.</td></tr>
<tr><td><b>3</b></td><td><b>Switch 1</b></td><td><b>Room Scenario</b></td><td><b>1 bit</b>
<b>DPT 1.001</b></td><td><b>C, W, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>Enable Room Scenarios 1…16,</i> the parameter Enable <i>Room Scenarios</i> has been selected with the option <i>yes</i>.
In accordance with the parameterization, this communication object can be set to ON/OFF or TOGGLE. With the setting <i>TOGGLE</i>, the value set beforehand, e.g. value 0 is toggled directly to the value 1 and vice versa.
Telegram value: 0 = OFF
1 = ON</td></tr>
<tr><td><b>4</b></td><td><b>Switch 2</b></td><td><b>Room Scenario</b></td><td></td><td></td></tr>
<tr><td colspan="5">See communication object No. 3</td></tr>
<tr><td><b>5</b></td><td><b>Automatic blind triggering</b></td><td><b>Room Scenario</b></td><td><b>1 bit</b>
<b>DPT 1.001</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>Enable Room Scenarios 1…16,</i> the parameter Enable <i>Room Scenarios</i> has been selected with the option <i>yes</i>.
Using this communication object, further KNX blind devices can be moved on automatic via the bus.
Telegram value: 0 = no activation of automatic blind
1 = activation of automatic blind</td></tr>
</table>

<table>
<tr><th>No.</th><th>Function</th><th>Object name</th><th>Data type</th><th>Flags</th></tr>
<tr><td><b>6</b></td><td><b>Recall KNX scene</b></td><td><b>Room Scenario</b></td><td><b>1 byte</b>
<b>DPT 18.001</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>Enable Room Scenarios 1…16,</i> the parameter Enable <i>Room Scenarios</i> has been selected with the option <i>yes</i>.
Using this 1 byte communication object, a scene recall can be sent using a coded scene telegram. The telegram contains the number of the respective scene as well as the information on whether the scene is to be retrieved, or if the current switch state is to be assigned to the scene.
Telegram format (1 byte): MXSSSSSS
(MSB) (LSB)
M: 0 – Scene is recalled
1 – store scene not possible
X: not used
S: Number of the scene (1…64: 00000000…00111111)
<table>
<tr><th colspan="2">1 byte telegram</th><th>Meaning</th></tr>
<tr><th>Decimal</th><th>Hexadecimal</th><th></th></tr>
<tr><td>00</td><td>00h</td><td>Recall scene 1</td></tr>
<tr><td>01</td><td>01h</td><td>Recall scene 2</td></tr>
<tr><td>02</td><td>02h</td><td>Recall scene 3</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
<tr><td>03</td><td>3Fh</td><td>Recall scene 64</td></tr>
</table></td></tr>
<tr><td><b>7</b></td><td><b>Trigger internal block</b></td><td><b>Room Scenario</b></td><td><b>1 bit</b>
<b>DPT 1.001</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>Enable Room Scenarios 1…16,</i> the parameter Enable <i>Room Scenarios</i> has been selected with the option <i>yes</i>.
KNX devices can be disabled via this communication object.
Telegram value: 0 = deactivate internal block.
1 = activate internal block.</td></tr>
<tr><td><b>8</b></td><td><b>Thermostat ON/OFF</b></td><td><b>Room Scenario</b></td><td><b>1 bit</b>
<b>DPT 1.001</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>Enable Room Scenarios 1…16,</i> the parameter Enable <i>Room Scenarios</i> has been selected with the option <i>yes</i>.
Telegram value: 0 = Thermostat OFF
1 = thermostat ON</td></tr>
<tr><td><b>9</b></td><td><b>Send value</b></td><td><b>Room Scenario</b></td><td><b>1 byte</b>
<b>DPT 5.010</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>Enable Room Scenarios 1…16,</i> the parameter Enable <i>Room Scenarios</i> has been selected with the option <i>yes</i>.
This communication object sends a value on the bus.
1 byte value [0…255]</td></tr>
</table>

3.3.4 Communication objects *Inputs a…h*

The communication objects of all Inputs do not differentiate from one another and are explained using *Input a*.

The descriptions of the parameter setting options of *Inputs a…h* are described in Parameter window Enable Inputs a…h, page 26.

The communication objects *Input a* have the nos. 10…14.

The communication objects *Input b* have the nos. 15…19.

The communication objects *Input c* have the nos. 20…24.

The communication objects *Input d* have the nos. 25…29.

The communication objects *Input e* have the nos. 30…34.

The communication objects *Input f* have the nos. 35…39.

The communication objects *Input g* have the nos. 40…44.

The communication objects *Input h* have the nos. 45…49.

3.3.4.1 Communication objects *Switch sensor*

<table>
<tr><th>No.</th><th>Function</th><th>Object name</th><th>Data type</th><th>Flags</th></tr>
<tr><td><b>10</b></td><td><b>Block</b></td><td><b>Input a: Switch sensor</b></td><td><b>1 bit</b>
<b>DPT 1.003</b></td><td><b>C, W</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>a: Switch sensor</i>, the parameter <i>Enable communication object “Block" 1 bit</i> has been selected with option <i>yes</i>.
Using the communication object <i>Block</i>, the input can be blocked or enabled. With activated communication object <i>Block</i> the inputs are blocked.

<b>Note</b>
When the input is blocked there is fundamentally no reaction to a signal change on the input, but:
– Waiting for a long button operation or a minimum signal duration is suspended.
– Parameterised <i>Cyclic sending</i> is not interrupted.
– The description of the communication object <i>Switch x</i> is still possible.
If the input state changed during the blocked phase, this leads to immediate sending of the new communication object value after enabling. If the input state remains the same during the blocking phase, the communication object value is not sent.

Telegram value 0 = enable input a
1 = block input a</td></tr>
<tr><td><b>11</b></td><td><b>Switch 1</b></td><td><b>Input a: Switch sensor</b></td><td><b>1 bit</b>
<b>DPT 1.001</b></td><td><b>C, W, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window <i>Enable inputs a…f</i> the parameter <i>Input a (binary input, contact scanning)</i> has been selected with the option <i>Switch sensor</i>.
In accordance with the parameter setting, this communication object can be switched by actuation of the input to <i>ON</i>, <i>OFF</i> or <i>TOGGLE</i> or can be set to <i>no reaction</i>. With toggle the previous value, e.g. 1, is toggled directly to the value 0. The communication object can be sent cyclically, e.g. for lifesign monitoring of the sensor.

<b>Note</b>
The communication object can be written to externally. Thus cyclic sending is interrupted or may not be possible depending on the parameter setting.
No further communication objects are visible with the setting.

Telegram value: 0 = OFF
1 = ON</td></tr>
<tr><td><b>12</b>
<b>13</b></td><td><b>Switch 2</b>
<b>Switch 3</b></td><td></td><td></td><td></td></tr>
<tr><td colspan="5">See communication object 11.</td></tr>
<tr><td><b>14</b></td><td><b>Start event 0/1</b></td><td><b>Input a: Switch sensor</b></td><td><b>1 bit</b>
<b>DPT 1.001</b></td><td><b>C, W</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>a: Switch sensor</i>, the parameter <i>Enable communication object “Start event 0/1" 1 bit</i> has been selected with option <i>yes</i>.
The 1 bit communication object <i>Start event 0/1</i> is enabled. As a result, the same events except those of the push button/switch connected to the binary input can also be triggered by the receipt of a telegram on the communication object <i>Event 0/1 started</i>.
Telegram value: 0 = start event 0
1 = start event 1</td></tr>
</table>

3.3.4.2 Communication objects *Switch/Dimming sensor*

<table>
<tr><td>No.</td><td>Function</td><td>Object name</td><td>Data type</td><td>Flags</td></tr>
<tr><td><b>10</b></td><td><b>Block</b></td><td><b>Input a:</b>
<b>Switch/dimming sensor</b></td><td><b>1 bit</b>
<b>DPT 1.003</b></td><td><b>C, W</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>a: Switch/Dimming sensor</i>, the parameter <i>Enable communication object “Block" 1 bit</i> has been selected with option <i>yes</i>.
Using the communication object <i>Block</i>, the input can be blocked or enabled. With activated communication object <i>Block</i> the inputs are blocked.
<b>Note</b>
When the input is blocked there is fundamentally no reaction to a signal change on the input, but:
– Waiting for a long button operation or a minimum signal duration is suspended.
– Parameterized <i>Cyclic sending</i> is interrupted with dimming steps.
– The description of the communication object <i>Switch</i> is still possible.
When enabling an input, a change of the signal states (as opposed to before the block) leads to immediate processing, e.g.:
– The minimum actuation or detection of a long/short button push starts.
– Communication objects send their value if necessary.
Telegram value 0 = enable input a
1 = block input a</td></tr>
<tr><td><b>11</b></td><td><b>Switch</b></td><td><b>Input a:</b>
<b>Switch/dimming sensor</b></td><td><b>1 bit</b>
<b>DPT 1.001</b></td><td><b>C, W, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window <i>Enable inputs a…f,</i> the parameter <i>Input a (binary input, contact scanning)</i> has been selected with the option <i>Switch/dim sensor</i>.
In accordance with the parameter setting, this communication object can be switched by actuation of the input to <i>ON</i>, <i>OFF</i> or <i>TOGGLE</i> or can be set to <i>no reaction</i>. With toggle the previous value, e.g. 1, is toggled directly to the value 0. With parameter setting <i>TOGGLE</i>, the communication object as the non-sending group address should be linked with the switch feedback of the dimming actuator (updating of the switching state).
<b>Note</b>
The communication object can be written to externally. Thus cyclic sending is interrupted or may not be possible depending on the parameter setting.
No further communication objects are visible with the setting.
Telegram value: 0 = OFF
1 = ON</td></tr>
<tr><td><b>12</b></td><td><b>Dimming</b></td><td><b>Input a:</b>
<b>Switch/dimming sensor</b></td><td><b>4 bit</b>
<b>DPT 3.007</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window <i>Enable inputs a…f,</i> the parameter <i>Input a (binary input, contact scanning)</i> has been selected with the option <i>Switch/dim sensor</i>.
A long operation at the input has the effect that BRIGHTER or DARKER dim telegrams are sent via this communication object on the bus. A STOP telegram is sent and the cyclic sending of dim telegrams is stopped at the end of actuation with START-STOP-DIMMING.</td></tr>
<tr><td><b>13, 14</b></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="5">Not assigned</td></tr>
</table>

3.3.4.3 Communication objects *Blind sensor*

<table>
<tr><th>No.</th><th>Function</th><th>Object name</th><th>Data type</th><th>Flags</th></tr>
<tr><td><b>10</b></td><td><b>Block</b></td><td><b>Input a:
Blind sensor</b></td><td><b>1 bit
DPT 1.003</b></td><td><b>C, W</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window <i>a: Blind sensor</i>, the parameter <i>Enable communication object “Block" 1 bit</i> has been selected with option <i>yes</i>.
Using the communication object <i>Block</i>, the input can be blocked or enabled. With activated communication object <i>Block</i> the inputs are blocked.

<b>Note</b>
When the input is blocked there is fundamentally no reaction to a signal change, but:
– Waiting for a long button operation or a minimum signal duration is suspended.
– Parameterized <i>Cyclic sending</i> is interrupted.
– Communication objects continue to be updated and sent if necessary.
When enabling an input, a change of the signal states (as opposed to before the block) leads to immediate processing, e.g.:
– The minimum actuation or detection of a long/short button push starts.
– Communication objects send their current value if necessary.

Telegram value 0 = enable input a
1 = block input a</td></tr>
<tr><td><b>11</b></td><td><b>Blind UP/DOWN</b></td><td><b>Input a:
Blind sensor</b></td><td><b>1 bit
DPT 1.008</b></td><td><b>C, W, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window <i>Enable inputs a…h</i> the parameter <i>Input a (binary input, contact scanning)</i> has been selected with the option <i>Blind sensor</i>.
This communication object sends a blind motion telegram UP or DOWN on the bus. By receiving telegrams, the device also recognises movement telegrams of another sensor, e.g. parallel operation.
Telegram value 0 = UP
1 = DOWN</td></tr>
<tr><td><b>12</b></td><td><b>STOP/slat adjustment</b></td><td><b>Input a:
Blind sensor</b></td><td><b>1 bit
DPT 1.007</b></td><td><b>C, T</b></td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window <i>Enable inputs a…h</i> the parameter <i>Input a (binary input, contact scanning)</i> has been selected with the option <i>Blind sensor</i>.
This communication object sends a STOP telegram or slat adjustment.
Telegram value 0 = STOP/slat adjustment OPEN
1 = STOP/slat adjustment CLOSE</td></tr>
</table>

No.	Function	Object name	Data type	Flags
13	**Upper limit position**	**Input a:** **Blind sensor**	**1 bit** **DPT 1.002**	**C, W**

This communication object is enabled if in the parameter window *Enable inputs a…h* the parameter *Input a (binary input, contact scanning)* has been selected with the option *Blind sensor*.

With this communication object, the feedback of a blind actuator, which indicates whether the blind is located in the upper end position, can be integrated.

Note
The communication object is important for 1-button operation (synchronisation).

Telegram value: 0 = blind is not in upper end position.
1 = blind has reached the upper end position.

No.	Function	Object name	Data type	Flags
14	**Lower limit position**	**Input a:** **Blind sensor**	**1 bit** **DPT 1.002**	**C, W**

This communication object is enabled if in the parameter window *Enable inputs a…h* the parameter *Input a (binary input, contact scanning)* has been selected with the option *Blind sensor*.

With this communication object, the feedback of a blind actuator which indicates whether the blind is located in the lower end position can be integrated.

Note
The communication object is important for 1-button operation (synchronisation).

Telegram value: 0 = blind is not in lower end position.
1 = blind has reached the lower end position.

3.3.4.4 Communication objects *Value/forced operation*

<table>
<tr><td>No.</td><td>Function</td><td>Object name</td><td>Data type</td><td>Flags</td></tr>
<tr><td>**10**</td><td>**Block**</td><td>**Input a:**
Value / forced operation</td><td>**1 bit**
DPT 1.003</td><td>**C, W**</td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window *a: Value/Forced operation,* the parameter *Enable communication object “Block" 1 bit* has been selected with option *yes*.
Using the communication object *Block*, the input can be blocked or enabled. With activated communication object *Block* the inputs are blocked.

Note
When the input is blocked there is fundamentally no reaction to a signal change, but:
– Waiting for a long button operation or a minimum signal duration is suspended.
– The parameter setting *8 bit scene* is ended with saving.
– Communication objects continue to be updated and sent if necessary.
When enabling an input, a change of the signal states (as opposed to before the block) leads to immediate processing, e.g.:
– The minimum actuation or detection of a long/short button push starts.
– Communication objects send their current value if necessary.

Telegram value 0 = enable input a
1 = block input a</td></tr>
<tr><td>**11**</td><td>**Value 1**</td><td>**Input a:**
Value / forced operation</td><td>**DPT variable**</td><td>**C, T**</td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window *Enable inputs a…f,* the parameter *Input a (binary input, contact scanning)* has been selected with the option *Value/forced operation*.
This communication object sends a value on the bus with short operation when opening or closing of the contact. The value and data type can be freely set in the parameters.

1 bit value [0/1] DPT 1.001 switch telegram
2 bit value [0…3] DPT 2.001 forced operation
1 byte value [-128…127] DPT 6.010 value
1 byte value [0…255] DPT 5.010 value
1 byte value [8 bit scene] DPT 18.001 control scene
2 byte value [-32,768...32,767] DPT 7.001 value

2 byte value [0...65,535] DPT 8.001 value
2 byte value [floating point] DPT 9.001 temperature
3 byte value [time of day, weekday] DPT 10.001 time of day, weekday
4 byte value [0…4294967295] DPT 12.001 value

4 byte value [-2147483648…2147483647] DPT 13.001 value</td></tr>
<tr><td>**12**</td><td>**Value 2**</td><td></td><td></td><td></td></tr>
<tr><td colspan="5">See communication object 11.</td></tr>
<tr><td>**13…14**</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="5">Not assigned</td></tr>
</table>

3.3.5 Communication objects *Outputs*

The communication objects of all outputs differentiate from one another with the exception of the communication objects *Logical connection 1* and *Logical connection 2.* . They are therefore explained using *Output A (6 A)*.

The descriptions of the parameter setting options of *Outputs A…D (6 A)* are described in Parameter window Enable Outputs A…H, page 57.

The communication objects *Output A (6 A)* have the nos. 50...57.

The communication objects *Output B (6 A)* have the nos. 58...65.

The communication objects *Output C (6 A)* have the nos. 66...73.

The communication objects *Output D (6 A)* have the nos. 74...81.

The communication objects *Output E (6 A)* have the nos. 82...89.

The communication objects *Output F (6 A)* have the nos. 90...97.

The communication objects *Output G (6 A)* have the nos. 98...105.

The communication objects *Output H (6 A)* have the nos. 106...113.

3.3.5.1 Communication objects *Output A (6 A)*

<table>
<tr><td>No.</td><td>Function</td><td>Object name</td><td>Data type</td><td>Flags</td></tr>
<tr><td>**50**</td><td>**Switch**</td><td>**Output A (6 A)**</td><td>**1 bit**
DPT 1.001</td><td>**C, W, T**</td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window *Enable Outputs A...D* the parameter *Output A (6 A)* has been enabled.
This communication object is used for switching of the output ON/OFF. The device receives a switch telegram via a switch communication object.
N/O:
Telegram value 1 = switch ON
0 = switch OFF
N/C:
Telegram value 1 = switch OFF
0 = switch ON
Note
With logical connections or forced operations, a modification of the communication object *Switch* does not necessarily lead to a change of the contact position.
For further information see: Function diagram, page 102.</td></tr>
<tr><td>**51**</td><td>**Permanent ON**</td><td>**Output A (6 A)**</td><td>**1 bit**
DPT 1.003</td><td>**C, W**</td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window *A: Output (6 A),* the parameter *Enable function Time* has been selected with the option *yes*.
The output can be forcibly switched on with this communication object.
If the communication object is assigned with the value 1, the output is switched on irrespective of the value of the object *Switch* and remains switched on until the communication object *Permanent ON* has the value 0. After ending the permanent ON state, the state of the communication object *Switch* is used.
Permanent ON only switches ON and “masks” the other functions. This means that the other functions, e.g. staircase lighting, continue to run in the background but do not initiate a switching action. After the end of permanent ON, the switching state, which would result without the permanent ON function, becomes active. For the function *Staircase lighting,* the response after permanent ON is parameterized in Parameter window A: Output (6 A) - Time, page 64.
This communication object can be used, for example, to allow the service or maintenance and cleaning personnel to initiate a permanent ON. The device receives a switch telegram via the switch object.
Permanent ON becomes inactive after a download or bus voltage recovery.
Telegram value 1 = activates permanent ON mode
0 = deactivates permanent ON mode</td></tr>
</table>

No.	Function	Object name	Data type	Flags
52	**Disable function time**	**Output A (6 A)**	**1 bit** **DPT 1.003**	**C, W**

This communication object is enabled if in parameter window *A: Output (6 A)*, the parameter *Enable function Time* has been selected with the option *yes*.

After bus voltage recovery, in parameter window - *Time* the communication object value with the parameter *Object value "Disable function Time"* can be determined.

With the blocked function *Time,* the output can only be switched ON or OFF, the function *Staircase lighting* is not triggered.

Telegram value 1 = staircase lighting disabled
0 = staircase lighting enabled

The contact position at the time of disabling and enabling is retained and will only be changed with the next switch telegram to the communication object *Switch*.

53	**Scene**	**Output A (6 A)**	**1 byte** **DPT 18.001**	**C, W**

This communication object is enabled if in parameter window *A: Output (6 A),* the parameter *Enable function Scene* has been selected with the option *yes*.

Using this 8 bit communication object, a scene telegram can be sent using a coded telegram. The telegram contains the number of the respective scene as well as the information on whether the scene is to be retrieved, or if the current switch state is to be assigned to the scene.

Telegram format (1 byte): MXSSSSSS
(MSB) (LSB)
M: 0 – Scene is recalled
1 – scene is stored (if allowed)
X: not used
S: Number of the scene (1…64: 00000000…00111111)

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

For further information see: Function Scene, page 109, and Code table scene (8 bit), page 124

54	**Forced operation**	**Output A (6 A)**	**1 bit** **DPT 1.003**	**C, W**

This communication object is enabled if in parameter window *A: Output (6 A)*, the parameter *Enable function forced operation* has been selected with the option *yes* and the parameter *Type of object "Forced operation"* has been selected with *1 bit.*

If the object receives the value 1, the output is forcibly set to the parameterized switch position, which has been set in the parameter window *Output A (6 A)*. The forced positioning of the contact should remain until forced operation is ended. This is then the case when a 0 is received via the communication object *Forced operation*.

Please note that the function *Forced operation* and a bus failure have a higher priority on the switching state, see Function diagram, page 102.

<table>
<tr><td>No.</td><td>Function</td><td>Object name</td><td>Data type</td><td>Flags</td></tr>
<tr><td>**54**</td><td>**Forced operation**</td><td>**Output A (6 A)**</td><td>**2 bit**
DPT 2.001</td><td>**C, W**</td></tr>
<tr><td colspan="5">This communication object is enabled if in parameter window *A: Output (6 A)*, the parameter *Enable function forced operation* has been selected with the option *yes* and the parameter *Type of object "Forced operation"* has been selected with *2 bit.*
The output can be forcibly operated via this communication object (e.g. by a higher-level control). The object value directly defines the forced position of the contact:
0 or 1 = The output is not forcibly operated.
2 = The output is forcibly switched off
3 = The output is forcibly switched on</td></tr>
<tr><td>**55**</td><td>**Status switch**</td><td>**Output A (6 A)**</td><td>**1 bit**
DPT 1.001</td><td>**C, R, T**</td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window *A: Output (6 A)*, the parameter *Enable communication object “Switch status” 1 bit* has been selected with *yes*.
You can parameterize whether the communication object value *no, update only, after a change* or *after a change or request* is sent on the bus. The communication object value directly indicates the current contact position of the switching relay.
The status value can be inverted.
Telegram value 1 = relay ON or OFF depending on the parameterization
0 = Relay OFF or ON depending on the parameterization</td></tr>
<tr><td>**56**</td><td>**Logical connection 1**</td><td>**Output A (6 A)**</td><td>**1 bit**
DPT 1.002</td><td>**C, W**</td></tr>
<tr><td colspan="5">This communication object is enabled if in the parameter window *- Logic* the parameters *Logical connection 1 active* has been selected with *yes*. The parameter window *- Logic* is enabled in the parameter window *A: Output A (6 A)*.
Using this communication object, the output of the first of two logic communication objects can be assigned. The logical connection is defined in the parameter window *- Logic*.
Initially, the switch communication object is logically linked with the communication object *Logical connection 1*. The result of this is then logically linked with the communication object *Logical connection 2*.
Note
The values of the communication objects *Logical connection 1/2* are stored at bus voltage failure. The values are set again after a bus voltage recovery
If values are not assigned for communication objects *Logical connection 1/2*, they will be deactivated.
At a reset via the bus, the values of the communication objects *Logical connection 1/2* remain unchanged.
For further information see: Connection/logic, page 107.</td></tr>
<tr><td>**57**</td><td>**Logical connection 2**</td><td>**Output A (6 A)**</td><td>**1 bit**
DPT 1.002</td><td>**C, W**</td></tr>
<tr><td colspan="5">See communication object 56.</td></tr>
</table>

4 Planning and application

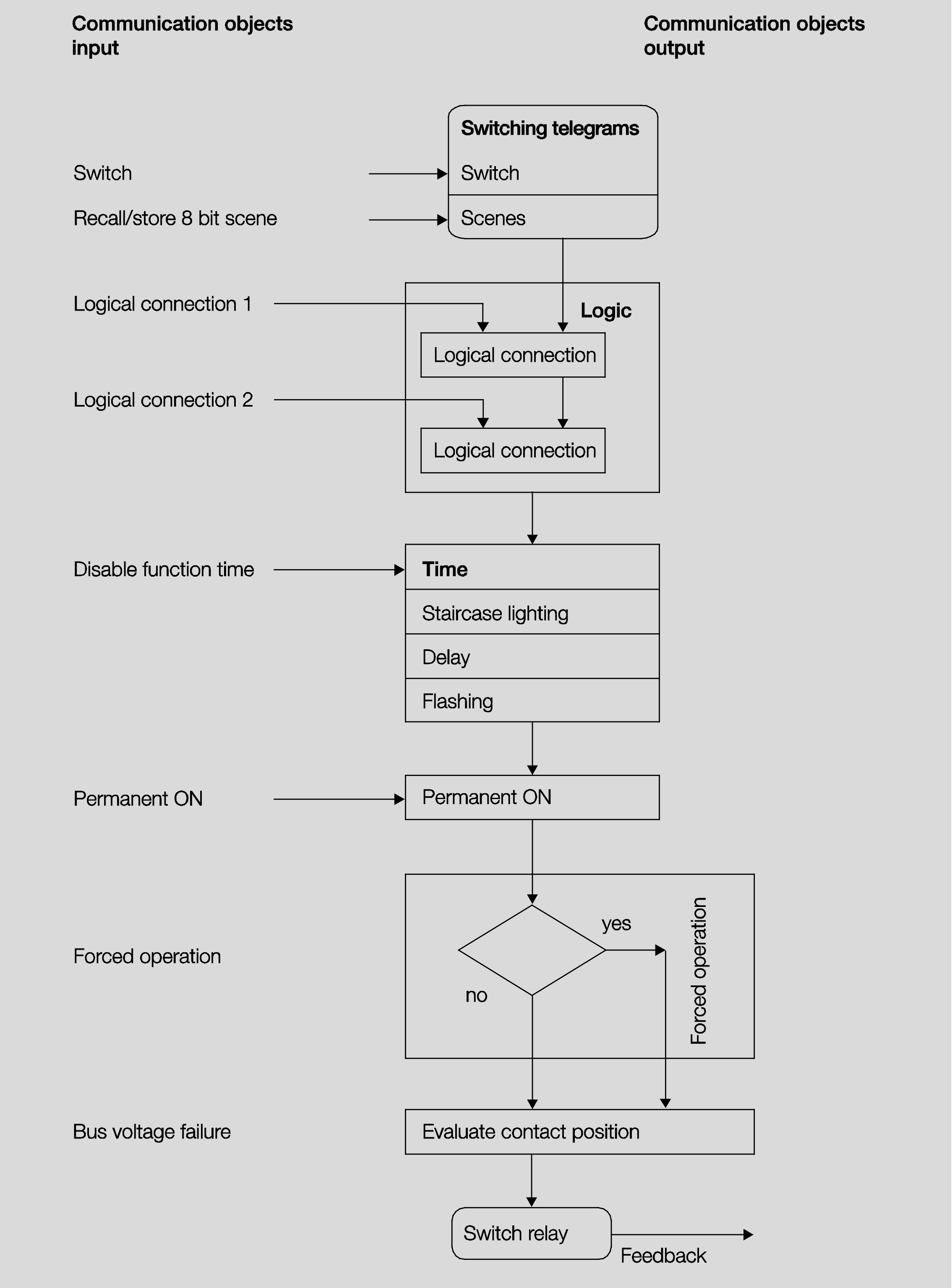
In this section you will find some tips and application examples for practical use of the device.

4.1 Output

In this chapter, the function charts and the application explanations for the outputs are explained.

4.1.1 Function diagram

The following illustration indicates the sequence, in which the functions are processed. Communication objects, which lead to the same box, have the same priority and are processed in the sequence, in which the telegrams are received.



Note
If a telegram is received via the communication object *Switch*, this is connected to both logical objects if they are activated. The result of this action serves as the input signal for the function *Time*. If this is not blocked, a corresponding switch signal is generated, e.g. delay or flashing. Before the switch telegram of the relay is reached, the forced operation is checked and executed as a priority if necessary. Subsequently, the switching action is only dependent on the state of the bus voltage. The relay is switched if a switching action allows it.

4.1.2 Function *Time*

The function *Time* can be enabled (value 0) and disabled (value 1) via the bus (1 bit communication object *Disable function time*). The output operates without a delay as long as the function *Time* is disabled.

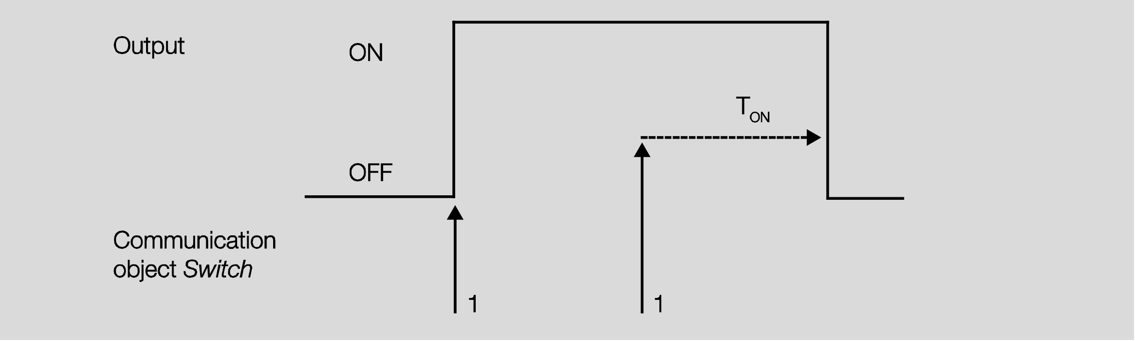
Different functions can be realised using the function *Time*:

- Staircase lighting
- Delay for switching ON and OFF
- Flashing

You can switch, for example, between functions, e.g. function *Staircase lighting* (night time operation) and normal ON/OFF switch function (daytime operation).

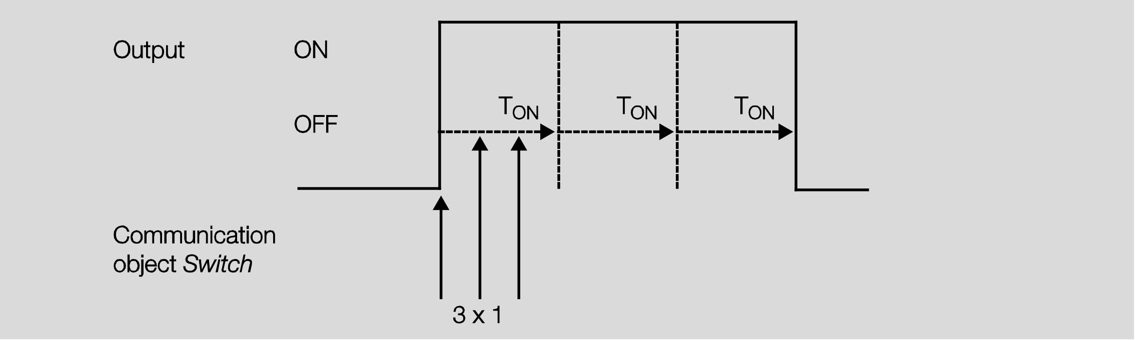
4.1.2.1 Staircase lighting

After the staircase lighting time T_{ON} the output switches off automatically. For every telegram with the value 1, the staircase lighting time restarts, unless the parameter *Extending staircase lighting by multiple operation ("pumping up")*, Parameter window A: Output (6 A) - Time, page 64, is set to *no (not retriggerable)*.



The response is the fundamental response of the staircase lighting function.

Via “pumping up” – actuation of the push button several times in succession – the user can adapt the staircase lighting to current needs. The maximum duration of the staircase lighting time can be set in the parameters.

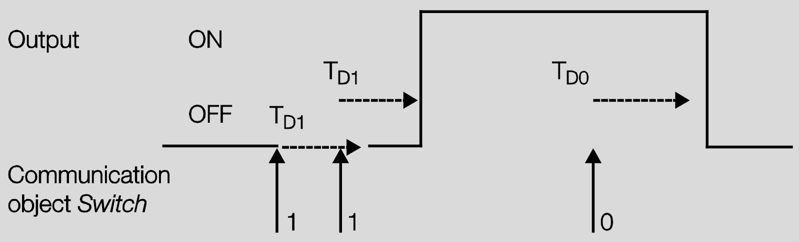


If the device receives a further ON telegram when the staircase lighting is switched on, the staircase lighting time is added to the remaining period.

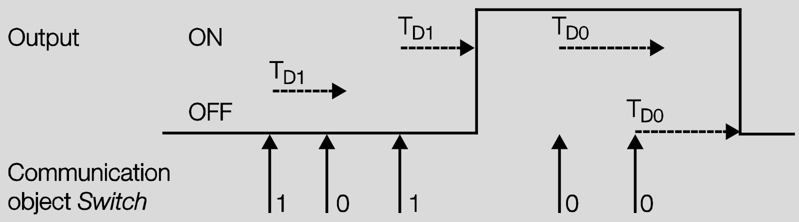
4.1.2.2 Delay for switching ON and OFF

The switching ON and OFF delay delays switch on or switch off of the output.

Example 1:



Example 2:



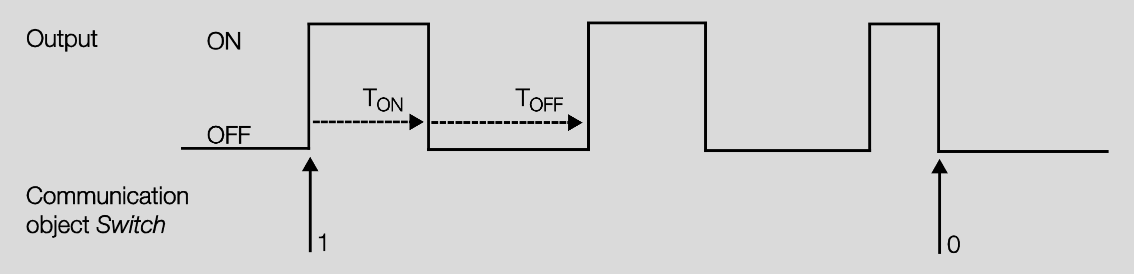
The delay time T_{D1} or T_{D0} starts after a switch telegram, and after it has timed out, the output executes the switch telegram.

If a new ON telegram with the value 1 is received during the switch on delay, the time of the switch on delay starts again. The same applies to switch off for the switch off delay. If a new OFF telegram with the value 0 is received during the switch off delay, the time of the switch off delay starts again.

Note
If the device receives an OFF telegram during the switch on delay T_{D1}, an ON telegram is disregarded.

4.1.2.3 Flashing

The output can flash when the output is switched on and off periodically.

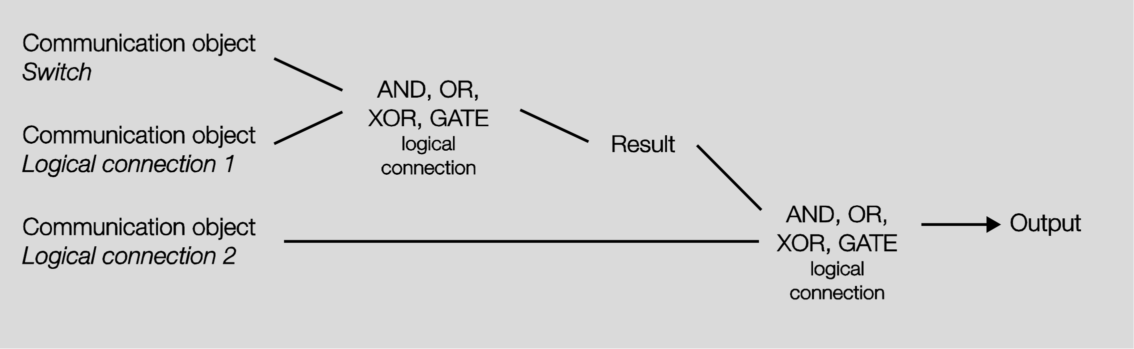


The switch on time (T_{ON}) and switch off time (T_{OFF}) during flashing can be programmed.

Note
The contact life of the contacts should be considered and can be found in the technical data. A limitation of the number of switching operations with the parameter *Number of impulses* may be useful. Furthermore, a delay in the switching sequence may possibly be caused by the limited availability of switching energy with very frequent switching. The possible number of switching operations should be considered.

4.1.3 Connection/logic

With the function *Connection/Logic,* it is possible to connect the switching of the output with certain conditions. Two connection communication objects are available:



At first, the communication object *Logical connection 1* is evaluated with the communication object *Switch*. The result is then logically linked with the communication object *Logical connection 2*.

The following logic functions are possible:

Communication object values						Explanations
Logical function	**Switch**	**Connection 1**	**Result**	**Connection 2**	**Output**	
AND	0	0	0	0	0	The result is 1 if both input values are 1. The output is 1 if both input values are 1.
	0	1	0	1	0	
	1	0	0	0	0	
	1	1	1	1	1	
OR	0	0	0	0	0	The result is 1 if one of both input values is 1.
	0	1	1	1	1	
	1	0	1	0	1	
	1	1	1	1	1	
XOR	0	0	0	0	0	The result is 1 when both input values have a different value.
	0	1	1	1	0	
	1	0	1	0	1	
	1	1	0	1	1	
GATE	0	closed		closed		The communication object *Switch* is only allowed through if the GATE (connection) is open. Otherwise the receipt of the communication object *Switch* is ignored.
	0	open	0	open	0	
	1	closed		closed		
	1	open	1	open	1	

The logic function is always re-calculated when a communication object value is received.

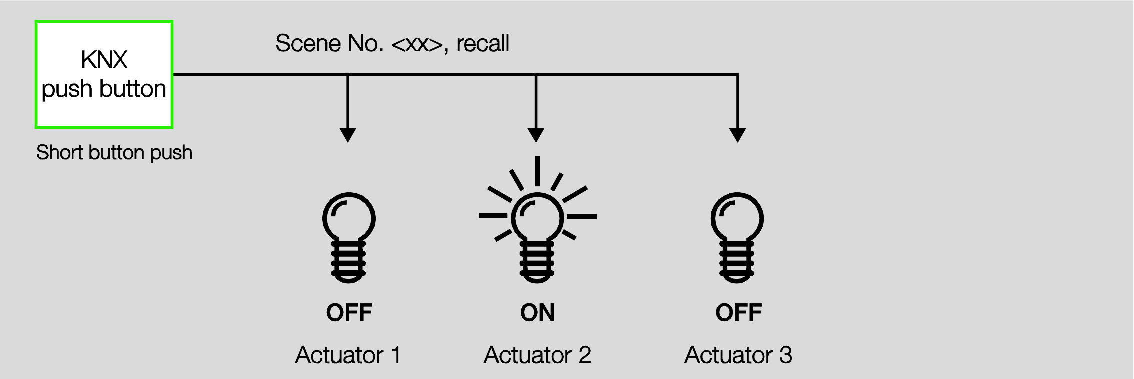
Gate function example
• The GATE logic is programmed, so that a disable is implemented as soon as the communication object *Logical connection x* receives a 0. • The output of the logical connection is 0. • The communication object *Logical connection 1* receives a 0, i.e. the GATE blocks. • The communication object *Switch* receives 0, 1, 0, 1. The output of the logic operation always remains 0. • The communication object *Logical connection x* receives a 1, i.e. the GATE is enabled if it is set in the parameters. • The output of the logical connection is recalculated.

Note
The values of the communication objects *Logical connection 1/2* are stored at bus voltage failure. The values are set again after a bus voltage recovery If values are not assigned for communication objects *Logical connection 1/2*, they will be deactivated. At a reset via the bus, the values of the communication objects *Logical connection 1/2* remain unchanged.

Note
If telegrams are received on the communication object *Switch* during the block, they will not be stored. For this reason, the output or the event remain unchanged when the GATE is enabled. The output switches if the GATE is enabled and a telegram is received on the communication object *Switch*.

4.1.4 Function *Scene*

With the scene using 8 bits, the push button issues the Room Master with the instruction to recall a scene. The scene is not stored in the push button, but rather in the Room Master.



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

Up to 64 different scenes are managed via a single group address. The scene telegram contains the recall or store functions of a scene.

In the following, the scene function is described which controls multiple KNX devices.

With the scene it is possible to retrieve one of 64 scenes or to connect multiple KNX devices in a scene. The scene can be retrieved or stored using a single telegram. It is a prerequisite that all the operating devices are parameterized with the same scene number.

Each KNX device involved receives the scene telegram and independently controls the scenes values. Using the Room Master, for example, the outputs are switched on or off, the blind moves to a determine position.

Up to 64 different scenes can be managed via a single KNX group address. The following information is contained in a scene telegram:

- Number of the scene (1…64)
- Recall scene / store scene

For further information see: Code table scene (8 bit), page 124.

Benefits
The function *Scene* with ABB i-bus® devices offers the following decisive advantage: All settings to be undertaken in a scene are stored in the device. Therefore, they must not be sent via the KNX when a scene is recalled, and only a figure value, which has been assigned to this scene, is necessary. This considerably reduces the load on the bus and prevents unnecessary telegram traffic on the KNX.

Note
The scene numbering 1 to 64 is retrieved via the KNX with a telegram number 0 to 63. For corresponding scene coding see Code table scene (8 bit), page 124.

4.2 Behaviour with …

4.2.1 Bus voltage recovery

General

- At bus voltage recovery, the communication object values can be parameterized; if not they are set to the value 0.
- Timers are out of operation and should be restarted.
- Status communication objects are sent as long as the option *after a change* has been set.
- The contact position is not known with 100 % certainty after bus voltage recovery. It is assumed that the contact position has not changed during the bus failure. Only after a new switch event is the contact position known to the Room Master.
- The send delay is only active at bus voltage recovery!

Switch contact output

- The communication object value *Staircase lighting time* remains unchanged as before bus voltage failure.
- The communication object value *Disable function time* is independent of the selected option.
- The communication object value *Permanent ON* remains unchanged as before bus voltage failure.
- The switch contact output switches as follows:
  - After the set communication object value *Switch* with bus voltage recovery.
  - If the parameter *Object value "Switch" at bus voltage recovery* is not parameterized, the behaviour at bus voltage failure is decisive.
  - If none of the two above options are selected, the last position is retained as with bus voltage failure.

Note
If a staircase lighting time was active at bus voltage failure, it will restart.

Note
The values of the communication objects *Logical connection 1/2* are stored at bus voltage failure. The values are set again after a bus voltage recovery If values are not assigned for communication objects *Logical connection 1/2*, they will be deactivated. At a reset via the bus, the values of the communication objects *Logical connection 1/2* remain unchanged.

Inputs

- The inactive waiting time is only active at bus voltage recovery.

4.2.2 ETS reset

What is an ETS reset?

Generally an ETS reset is defined as a reset of the device via the ETS. The ETS reset is initiated in the ETS3 under the menu item *Commissioning* with the function *Reset device*. This stops the application and it is restarted.

Note
For all resets after delivery including the first download, the response will comply with that of a reset via the bus. A send and switch delay is not executed. All states are reset.

Switch contact output

- The communication object value *Staircase lighting time* receives its parameterized value.
- The communication object value *Disable function time* is 0, i.e., function *Time* is not blocked.
- The object value *Permanent ON* is 0, i.e., permanent on is not active.
- The switch contact output goes to the safely opened state.

Note
The values of the communication objects *Logical connection 1/2* are stored at bus voltage failure. The values are set again after a bus voltage recovery If values are not assigned for communication objects *Logical connection 1/2*, they will be deactivated. At a reset via the bus, the values of the communication objects *Logical connection 1/2* remain unchanged.

4.2.3 Download (DL)

Note
After a download with a change, the application complies in behaviour to a reset of the device in the ETS.

Switch contact output

The communication object value *Staircase lighting time* remains unchanged.

The communication object value *Disable function time* remains unchanged.

> **Exception:** The communication object value is set to 0 if there is no assignment to the communication object.
>
> | **Note** |
> |---|
> | Otherwise, the block for the function *Time* is removed, if the communication object *Disable function time* is not available.
The switch contact output will otherwise use the new parameters. |

The communication object value *Permanent ON* remains unchanged.

The switch contact output remains unchanged.

4.2.4 Reaction on bus voltage failure

After the contact positions have set with bus voltage recovery, the device remains functional until the bus voltage recovers.

Note
The values of the communication objects *Logical connection 1/2* are stored at bus voltage failure. The values are set again after a bus voltage recovery If values are not assigned for communication objects *Logical connection 1/2*, they will be deactivated. At a reset via the bus, the values of the communication objects *Logical connection 1/2* remain unchanged.

Only the energy for a non-delayed switching action for each output is available should the bus voltage fail.

5 Room Scenarios

In this chapter, the method of function of the Room Scenarios is described.

5.1 Triggering Room Scenarios

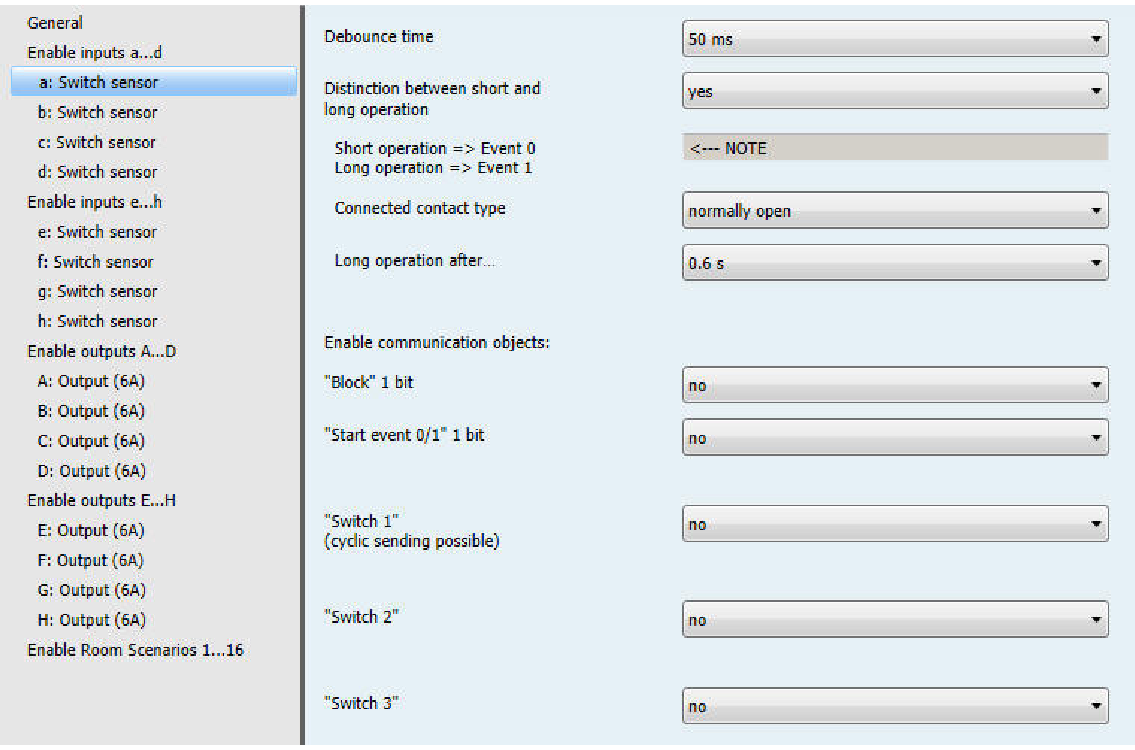
A Room Scenario consists of two events. Thereby, one event will trigger up to seven telegrams immediately, and the other event can trigger the same seven telegrams via a delay set with a timer.

Each of these events can be parameterized individually:

- Sending of two 1 bit values,
- Activation of the automatic function of a blind,
- Triggering a KNX scene, internally or via the bus,
- Deactivation/activation of the internal block of the binary inputs,
- Switching on/off the thermostat, e.g. RDF/A,
- Send 1 byte value

5.1.1 Room Scenario internal triggering

Every binary input can be triggered by two Room Scenarios linked to one another. The binary value 0 always triggers a room scenario with odd numbering, i.e. 1, 3, 5, 7, 9, 11, 13 or 15, and binary value 1 triggers a room scenario with even numbering, i.e. 2, 4, 6, 8, 10, 12, 14 or 16.



Debounce time: 50 ms

Distinction between short and long operation: yes

Short operation => Event 0
Long operation => Event 1: <--- NOTE

Connected contact type: normally open

Long operation after...: 0.6 s

Enable communication objects:

"Block" 1 bit: no

"Start event 0/1" 1 bit: no

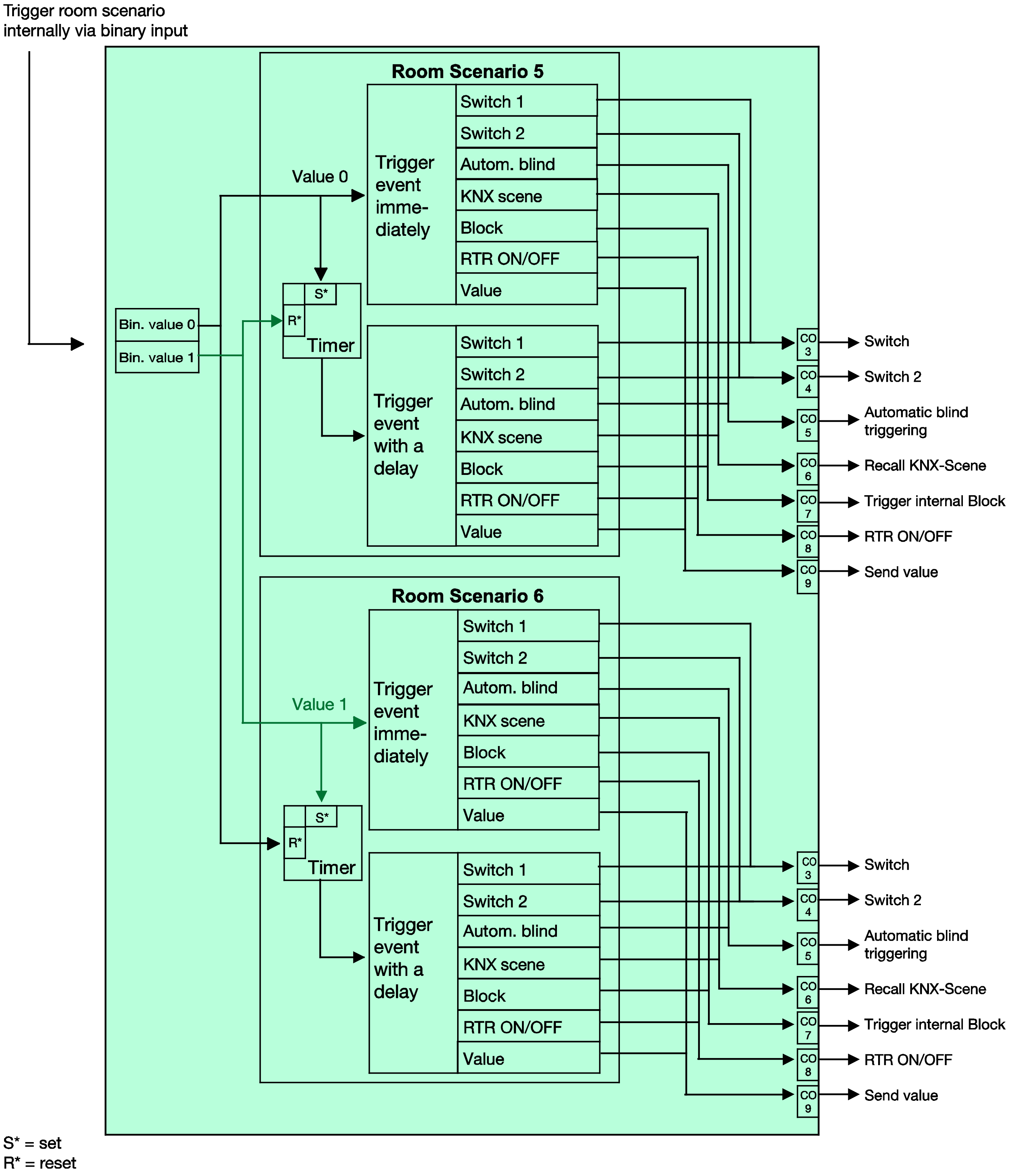
"Switch 1"
(cyclic sending possible): no

"Switch 2": no

"Switch 3": no

However, only one Room Scenario can be active at a time. An activated Room Scenario can however trigger two events, one of them immediately and the other delayed via a timer. Through the connection in pairs of the Room Scenarios, the binary value triggers one of both connected Room Scenarios and overwrites the previous Room Scenario.

The following overview shows the method of function based on Room Scenarios 5 and 6:



S* = set
R* = reset

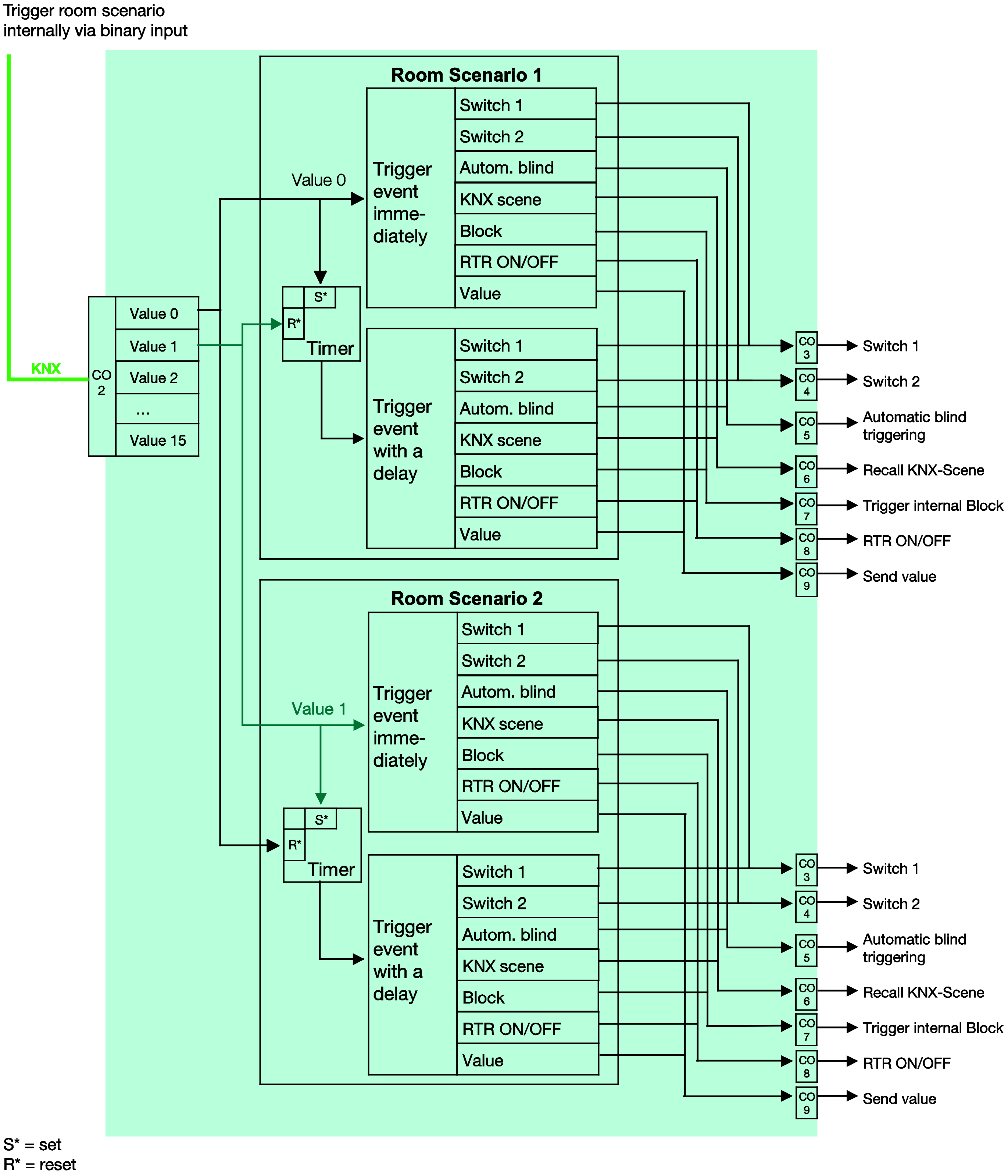
5.1.2 Room Scenario external triggering

A Room Scenario can also be triggered externally via the bus by the receipt of a 1 byte value on the communication object no. 2. The 1 byte values are divided as follows:

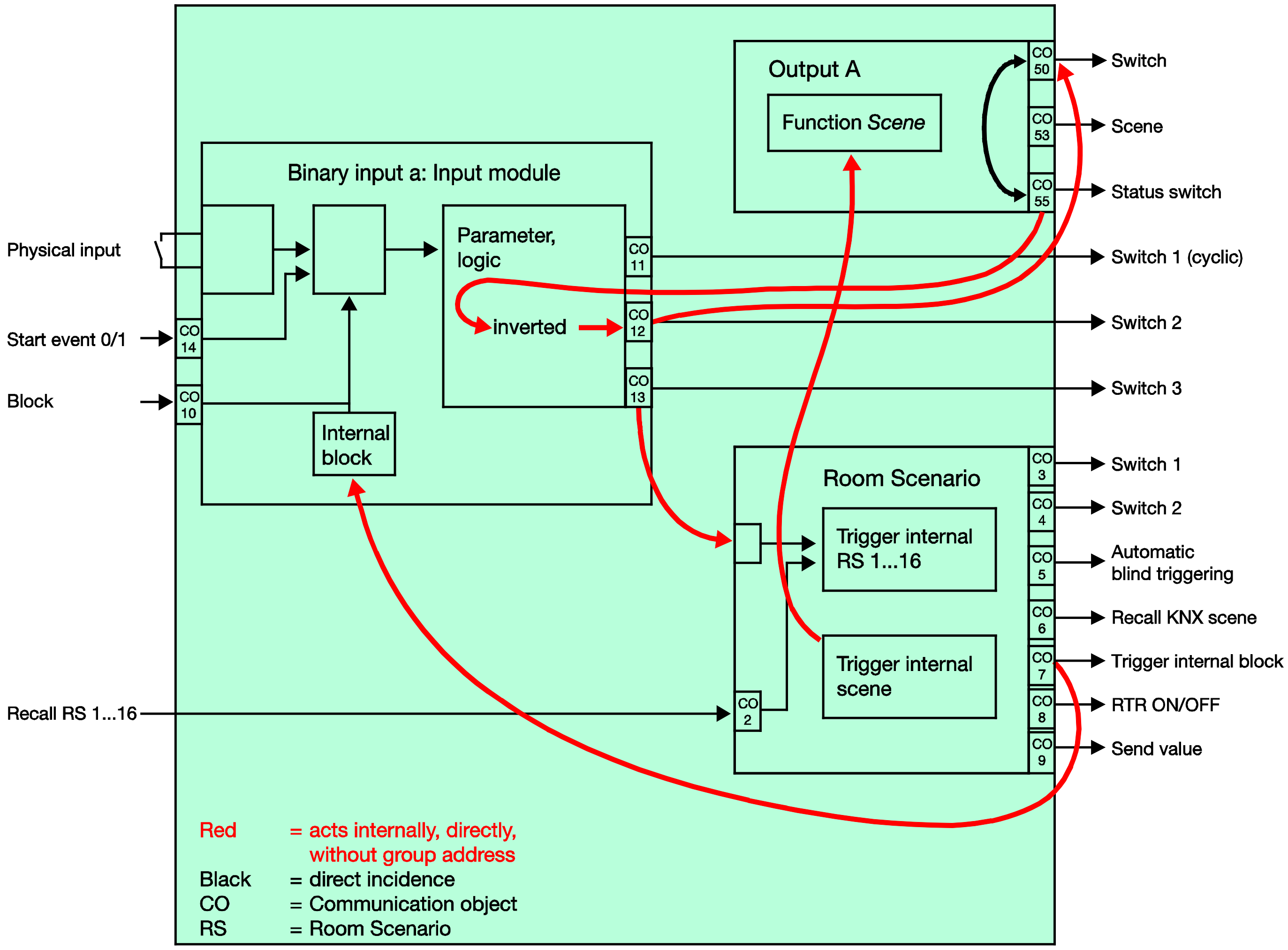
0 = Room Scenario 1
1 = Room Scenario 2
2 = Room Scenario 3
3 = Room Scenario 4
4 = Room Scenario 5
5 = Room Scenario 6
6 = Room Scenario 7
7 = Room Scenario 8
8 = Room Scenario 9
9 = Room Scenario 10
10 = Room Scenario 11
11 = Room Scenario 12
12 = Room Scenario 13
13 = Room Scenario 14
14 = Room Scenario 15
15 = Room Scenario 16

The 1 byte values 16…255 are not occupied.

The following overview shows the method of function based on Room Scenarios 1 and 2:



5.2 Special feature Switch sensor



Note

This block diagram is only valid if a binary input is parameterized as a switch sensor with the *TOGGLE* switch function.

Parameterization of the binary input a:

Switch sensor

Switch 1: not used
Switch 2: switches directly on output A (6 A) *TOGGLE*
Switch 3: activates a Room Scenario

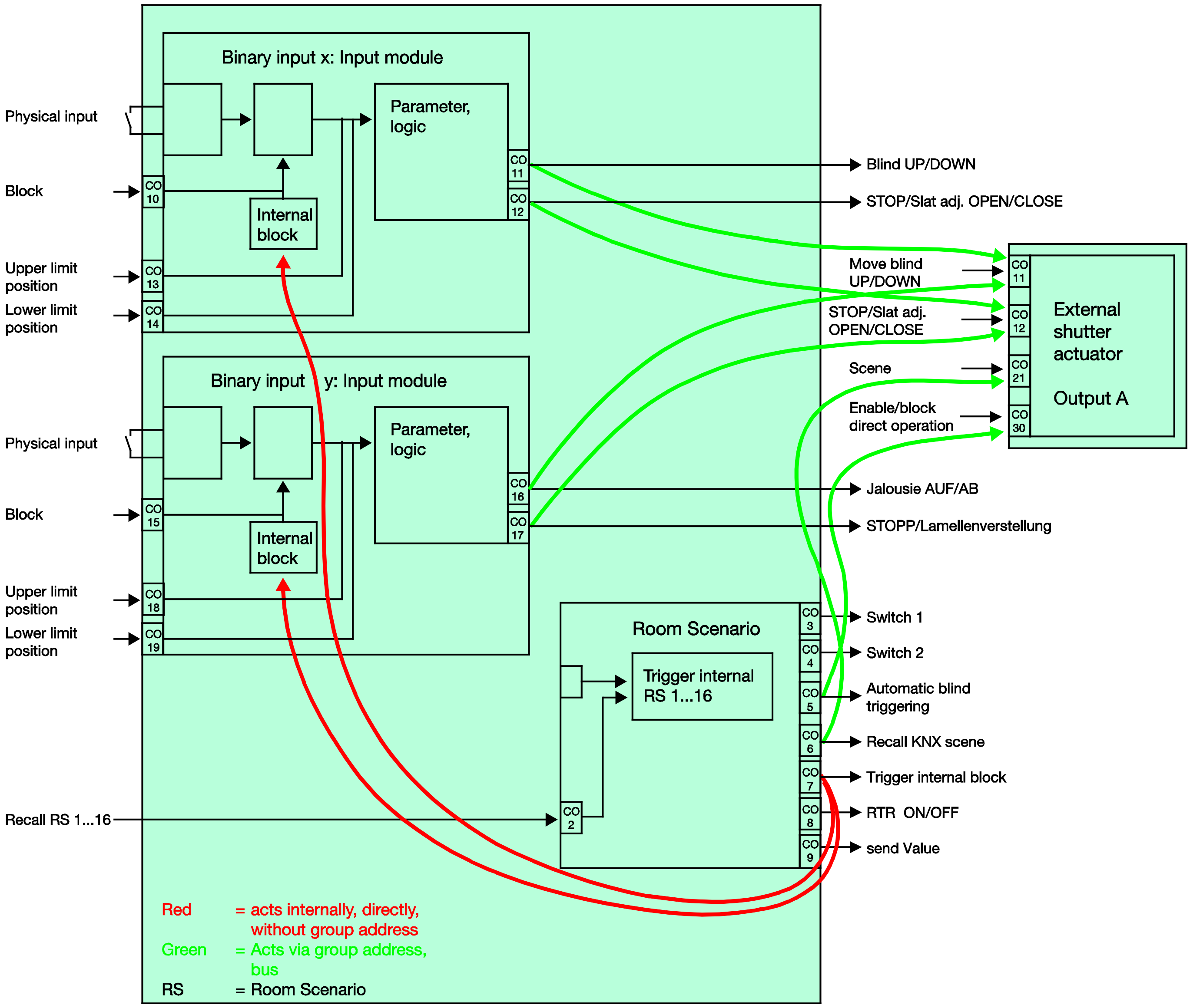
Parameterization of the output A (6 A):

N/O

Enable communication object no. 35: "Status Switch" 1 bit = yes
Send object value = no, update only
Object value of contact position: 1 = closed, 0 = open
Enable function scene = yes

Note
The parametric programming as a N/O contact and the contact position must be matched to one another to ensure that the status of the output is correctly fed back to communication object *Switch 2*. In this way, pressing a button twice for ON/OFF switching is prevented.

5.3 Special feature blind sensor with external blind actuator



Parameterization of the binary input x:

2 button operation

Short operation = STOP/slat OPEN
Long operation = move UP

Parameterization of the binary input y:

2 button operation

Short operation = STOP/slat CLOSE
Long operation = move DOWN

A Appendix

A.1 Scope of delivery

The Room Master is supplied together with the following components. The delivered items should be checked according to the following list.

- 1 pc. RM/S 4.1, Room Master Standard, MDRC
- 1 pc. Installation and operating instructions
- 1 pc. Bus connection terminal (red/black)

A.2 Code table scene (8 bit)

Bit No.		7	6	5	4	3	2	1	0		
8 bit value	Hexadecimal	Recall	Not defined	Scene number	Scene number	Scene number	Scene number	Scene number	Scene number	Scene number	Recall (R)
0	00									1	R
1	01								■	2	R
2	02							■		3	R
3	03							■	■	4	R
4	04						■			5	R
5	05						■		■	6	R
6	06						■	■		7	R
7	07						■	■	■	8	R
8	08					■				9	R
9	09					■			■	10	R
10	0A					■		■		11	R
11	0B					■		■	■	12	R
12	0C					■	■			13	R
13	0D					■	■		■	14	R
14	0E					■	■	■		15	R
15	0F					■	■	■	■	16	R
16	10				■					17	R
17	11				■				■	18	R
18	12				■			■		19	R
19	13				■			■	■	20	R
20	14				■		■			21	R
21	15				■		■		■	22	R
22	16				■		■	■		23	R
23	17				■		■	■	■	24	R
24	18				■	■				25	R
25	19				■	■			■	26	R
26	1A				■	■		■		27	R
27	1B				■	■		■	■	28	R
28	1C				■	■	■			29	R
29	1D				■	■	■		■	30	R
30	1E				■	■	■	■		31	R
31	1F				■	■	■	■	■	32	R
32	20			■						33	R
33	21			■					■	34	R
34	22			■				■		35	R
35	23			■				■	■	36	R
36	24			■			■			37	R
37	25			■			■		■	38	R
38	26			■			■	■		39	R
39	27			■			■	■	■	40	R
40	28			■		■				41	R
41	29			■		■			■	42	R
42	2A			■		■		■		43	R
43	2B			■		■		■	■	44	R
44	2C			■		■	■			45	R
45	2D			■		■	■		■	46	R
46	2E			■		■	■	■		47	R
47	2F			■		■	■	■	■	48	R
48	30			■	■					49	R
49	31			■	■				■	50	R
50	32			■	■			■		51	R
51	33			■	■			■	■	52	R
52	34			■	■		■			53	R
53	35			■	■		■		■	54	R
54	36			■	■		■	■		55	R
55	37			■	■		■	■	■	56	R
56	38			■	■	■				57	R
57	39			■	■	■			■	58	R
58	3A			■	■	■		■		59	R
59	3B			■	■	■		■	■	60	R
60	3C			■	■	■	■			61	R
61	3D			■	■	■	■		■	62	R
62	3E			■	■	■	■	■		63	R
63	3F			■	■	■	■	■	■	64	R

Bit No.		7	6	5	4	3	2	1	0		
8 bit value	Hexadecimal	Save	Not defined	Scene number	Scene number	Scene number	Scene number	Scene number	Scene number	Scene number	Save (S)
128	80	■								1	S
129	81	■							■	2	S
130	82	■						■		3	S
131	83	■						■	■	4	S
132	84	■					■			5	S
133	85	■					■		■	6	S
134	86	■					■	■		7	S
135	87	■					■	■	■	8	S
136	88	■				■				9	S
137	89	■				■			■	10	S
138	8A	■				■		■		11	S
139	8B	■				■		■	■	12	S
140	8C	■				■	■			13	S
141	8D	■				■	■		■	14	S
142	8E	■				■	■	■		15	S
143	8F	■				■	■	■	■	16	S
144	90	■			■					17	S
145	91	■			■				■	18	S
146	92	■			■			■		19	S
147	93	■			■			■	■	20	S
148	94	■			■		■			21	S
149	95	■			■		■		■	22	S
150	96	■			■		■	■		23	S
151	97	■			■		■	■	■	24	S
152	98	■			■	■				25	S
153	99	■			■	■			■	26	S
154	9A	■			■	■		■		27	S
155	9B	■			■	■		■	■	28	S
156	9C	■			■	■	■			29	S
157	9D	■			■	■	■		■	30	S
158	9E	■			■	■	■	■		31	S
159	9F	■			■	■	■	■	■	32	S
160	A0	■		■						33	S
161	A1	■		■					■	34	S
162	A2	■		■				■		35	S
163	A3	■		■				■	■	36	S
164	A4	■		■			■			37	S
165	A5	■		■			■		■	38	S
166	A6	■		■			■	■		39	S
167	A7	■		■			■	■	■	40	S
168	A8	■		■		■				41	S
169	A9	■		■		■			■	42	S
170	AA	■		■		■		■		43	S
171	DOW	■		■		■		■	■	44	S
172	AC	■		■		■	■			45	S
173	AD	■		■		■	■		■	46	S
174	AE	■		■		■	■	■		47	S
175	AF	■		■		■	■	■	■	48	S
176	B0	■		■	■					49	S
177	B1	■		■	■				■	50	S
178	B2	■		■	■			■		51	S
179	B3	■		■	■			■	■	52	S
180	B4	■		■	■		■			53	S
181	B5	■		■	■		■		■	54	S
182	B6	■		■	■		■	■		55	S
183	B7	■		■	■		■	■	■	56	S
184	B8	■		■	■	■				57	S
185	B9	■		■	■	■			■	58	S
186	BA	■		■	■	■		■		59	S
187	BB	■		■	■	■		■	■	60	S
188	BC	■		■	■	■	■			61	S
189	BD	■		■	■	■	■		■	62	S
190	BE	■		■	■	■	■	■		63	S
191	BF	■		■	■	■	■	■	■	64	S

empty = value 0

■ = value 1, applicable

Note

All combinations not listed or indicated are invalid.

A.3 Input 4 bit dimming telegram

The following table describes the 4 bit dim telegram:

Dec.	Hex.	Binary	Dim telegram
0	0	0000	STOP
1	1	0001	100 % DARKER
2	2	0010	50 % DARKER
3	3	0011	25 % DARKER
4	4	0100	12.5 % DARKER
5	5	0101	6.25 % DARKER
6	6	0110	3.13 % DARKER
7	7	0111	1.56 % DARKER
8	8	1000	STOP
9	9	1001	100 % BRIGHTER
10	R	1010	50 % BRIGHTER
11	B	1011	25 % BRIGHTER
12	C	1100	12.5 % BRIGHTER
13	D	1101	6.25 % BRIGHTER
14	E	1110	3.13 % HELLER
15	F	1111	1.56 % BRIGHTER

A.4 Ordering information

Short description	Description	Order No.	bbn 40 16779 EAN	Price group	Weight 1 pc. [kg]	Pack unit [pc.]
RM/S 4.1	Room Master, MDRC	2CDG 110 170 R0011	88126 5	P2	0.55	1

Notes

Notes

Notes

Contact us

ABB STOTZ-KONTAKT GmbH
Eppelheimer Straße 82
69123 Heidelberg, Germany
Phone: +49 (0)6221 701 607
+49 (0)6221 701 434 (KNX Helpline)
Fax: +49 (0)6221 701 724
E-mail: knx.marketing@de.abb.com
knx.helpline@de.abb.com

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

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

We reserve the rights to this document and all the items and illustrations contained therein. Reproduction, transfer to third parties or processing of the content – including sections thereof – is not permitted without prior expressed written permission from ABB AG.

Publication Number 2CDC 514 070 D0201 (12.12)

Power and productivity
for a better world™