B.E.G. LUXOMAT® net

KNX Switching actuator
Application description

Table of contents

Index

Part A - General

1. Introduction

The two B.E.G. switch actuators SA4-230/16/KNX REG and SA4-230/16/EM KNX REG receive and send KNX telegrams and switch 4 connected loads independently of each other. Each output, also known as a Channel, is switched by a relay. Each output is individually programmable via ETS. There is a choice of logical connections, status reports, blocking, central switching and many time functions, e.g. on/off delays and stairway control functions, as well as a blink function. Scene functions are also available.

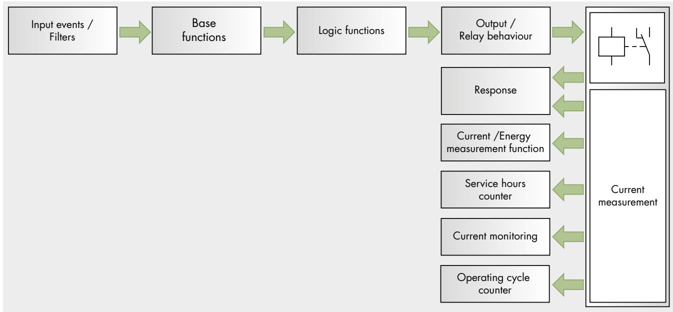
In addition, the SA4-230/16/EM KNX REG switch actuator has transformer-based current measurement (± 10mA), true effective value measurement (current) and voltage-synchronous Active power measurement.

1.1 General functions of the switch actuator

The switch actuator has two areas of functionality, switching and current measurement. The Base function of the switch actuator, switching, is carried out via four consecutive blocks, in which each event is processed:

- *Input events / Filters*
  An input event is a button press, for example. In this block, this input event can be filtered or inverted according to the object values set up for the block. The result obtained in this block is output and serves as the input event in the next block. The next block is the
- *Base functions*
  The Base functions available to the switch actuator are switching, stairway control and blinking. Parameters can also be set for these functions. The following block is the
- *Logic functions*
  The type of Logical connection can be selected here. In addition, the subsequent functions in the priority list (higher priority functions) of Blocking, Forced operation and Safety can be defined. Next, the
- Output / relay behaviour can be defined. In particular, the type of contact (NC/NO) can be established and response behaviour can be set in the parameters. The result produced here then determines switching behaviour.

Supplementary to the switching function, the load current for each channel can be measured when the relay is closed. Using the relay state and the measured current, the results of the current/energy measurement function, current monitoring and service hours counter are derived.



2. Base settings

The basic switch actuator functions are defined in base settings.

2.1 Start-up delay

There are often many actuators in a system. To avoid spikes when power is restored, each switch actuator can have a start-up delay assigned. This is the duration after bus voltage recovery that the switch actuator should wait before it returns to duty.

Base settings	
Start-up behaviour in seconds	0 – 120 **[5]**

2.2 In-service telegram (heartbeat)

When the switch actuator is ready for service, it can send an “in-service telegram” at intervals. This telegram is monitored by a subsequent function.
The telegram says only that the switch actuator is ready for service in itself. If a channel is defective, e.g. because a relay is “stuck”, this is not reported. The intervals at which this in-service telegram is sent can also be defined.

Base settings	
In-service telegram (heartbeat)	**deactivated**
	activated
Cycle time in minutes	1 – 120 **[60]**

2.3 Combined response

There are two options for responses. With active response (active response object) the relay state is reported to the bus at each change. With passive status objects, there is no automatic sending of the value. The object value is always current, but must be read off via the bus, e.g. by visualisation software. Here, it can be defined that responses from individual channels should be combined. Further information can be found in the “Responses” section.

Combined response	**active response object**
	passive status object

No.	Name	Function	C	R	W	T	U
242	General output (DPT 27.001)	Combined response (active)	C	–	–	T	–
242	General output (DPT 27.001)	Combined response (passive)	C	R	–	–	–

2.4 Reset actuator to original ETS parameters (reset)
The option exists to change some parameters while in service via bus access (objects). A learnt value can be protected against change (ETS download or actuator reset). In order to generally prevent changed parameters from being reset, this function must be deactivated. If the function is activated, all parameters for which a Reset is allowed, are reset.

A “1” telegram to the “Parameter Reset” object resets the actuator to the original ETS values. The values to be reset can be selected for each actuator function.

The parameters also have an influence on the next ETS download.
A learnt value can be protected against change (ETS download or actuator reset).

The table below shows which functions can be reset by the “Parameter Reset” object (“1” telegram).

Function	Parameter	Reset value
Scene function	Stored scenes by ETS download or object Reset	Value of input event for Scenes A to H
Delayed switch-on	Times changed by object by ETS download or object Reset	Delay time (hours, minutes, seconds)
Delayed switch-off	Times changed by object by ETS download or object Reset	Delay time (hours, minutes, seconds)
Stairway controller	Times changed by object by ETS download or object Reset	Follow-up time (hours, minutes, seconds)
Current / energy	Reset current energy value by ETS download or object Reset	0
Adaptive monitoring	Learnt current value by ETS download or object Reset	Current value in milliamps
Adaptive monitoring	Learnt active power by ETS download or object Reset	Active power in Watts
Service hours counter	**Limit changed** by object by ETS download or object Reset	Service hours limit in hours
Service hours counter	**Current service hours status** resettable by ETS download, object Reset	0
Operating cycle counter	**Limit changed** by object by ETS download or object Reset	Operating cycle counter limit
Operating cycle counter	**Current operating cycle counter status** resettable by ETS download, object Reset	0

Base settings	
Reset actuator to original download parameters	activated
	deactivated

No.	Name	Function	C	R	W	T	U
241	R1: Input (DPT 1.015)	Parameter reset	C	–	W	–	–

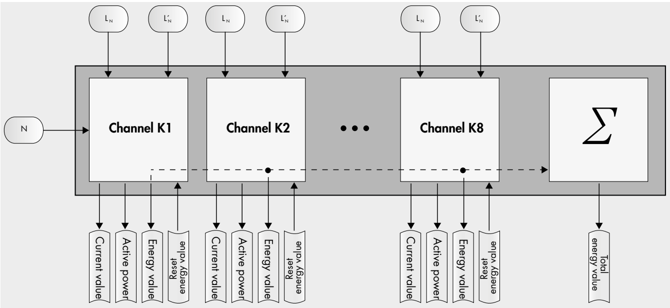
2.5 Total energy value
The switch actuator offers the option of calculating the total of the channels‘ individual energy values. For this, energy calculation should be selected for the channels to be included in the calculation (parameter “Current measurement / Energy calculation” = “activated”).

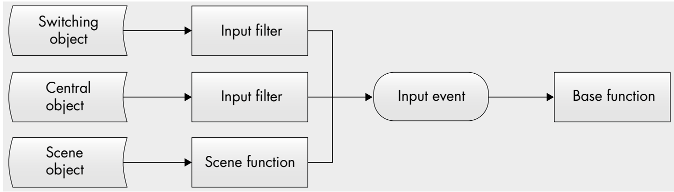
If a channel’s energy value is reset, this is taken into account in the calculation.

The total can be set up to be available as a status, i.e. only sent to the bus on request (e.g. visualisation). Alternatively, it can be set to send at intervals and/or when there is a change.
Please see the section “Current / Energy”.

Base settings	
Delay in evaluation after relay closing in seconds	0 – 60 **[10]**
Total energy value	**Status**
	send at intervals
	send on change
	send at intervals and on change
Hours (visible if sending at intervals)	0 – 24 **[10]**
Minutes (visible if sending at intervals)	0 – 59 **[0]**
Send on change by (visible if sending on change)	1 kWh
	5 kWh
	10 kWh

No.	Name	Function	C	R	W	T	U
244	General: Output (DPT 13.013)	Reporting of total energy value (in kWh)	C	–	–	T	–





Part B - Switching / Channel 1 to 8
Using the "Channel selection" parameter, Channels 1 to 8 can be activated or deactivated individually.

Channel selection	
Channel 1	**activated**
	deactivated
Channel 2	**activated**
	deactivated
Channel 3	**activated**
	deactivated
Channel 4	**activated**
	deactivated

The activated channels can then be set up individually. The functions available are the same for all channels. Below, the functions for one channel are explained as an example.

3. Input events / Filters
Each channel is assigned a Base function. There are three Base functions: Switching mode, Stairway control and Blink function. The Base functions are mutually exclusive, i.e. exactly one of these functions can be assigned to a channel.

Input objects for a channel are the switching object, the central object and the scene object. The switching object can trigger different reactions according to the parameters set up and can, for example, control a stairway installation or activate the Blink function. An input event is assigned to a Base function via an input filter.

The purpose of scene objects is to call stored scenes. For this purpose, a scene number (1 – 64) is sent over the KNX bus. Each channel can be assigned 8 scene numbers. Each scene number can only be assigned one input event, which then applies to the Base function. For scenes, the event is defined in the function, and therefore not filtered.

3.1 Input objects: Switching and central object
Each channel is assigned a switching object. An input event triggers a telegram, whose value can be set in the parameters. For example, the input event triggers switching or the start of the stairway controller.

The central object is a 1-bit object. This object can act on all channels. Whether the channel should evaluate the central object or not can be determined on a per channel basis. A telegram to this object is to be equated with telegrams to the switching object. In any case, the central function has its own input filter.

Input objects all have the same priority, i.e. the last telegram always prevails.

R1: Input events / Filters *(visible if Channel 1 is activated)*	
Value of input event if switching object = "1"	"0"
	"1"
	no reaction
Value of input event if switching object = "0"	**"0"**
	"1"
	no reaction
Scene function	**deactivated**
	activated
Value of input event if central object = "1"	"0"
	"1"
	no reaction
Value of input event if central object = "0"	"0"
	"1"
	no reaction

No.	Name	Function	C	R	W	T	U
0	R1: Input (DPT 1.001)	Switching	C	–	W	–	–
240	General: Input (DPT 1.001)	Central switching	C	–	W	–	–

Notes:
- The input result determined by the input filter is not sent directly to the relay. Only the Base function and the result from the logic block produce the switching state.
- The filter setting "No reaction" enables functionality across objects. For example, it is possible to switch on a channel with only a switching object. Switching off then takes place via a central object.

3.2 Scene function
For each channel, 8 independent scenes can be stored. Once the scene function has been activated for a channel, the scene object appears. If this is described with a scene number (1 to 64), the corresponding scene is called.

The current channel/relay state can be stored as a new scene. This also applies if the relay state is implemented by a high priority logical connection. In any case, only the relay state is implemented in the scene, and not in fact the higher priority logical connection.

The input result affects the downstream Base function, i.e. a scene can also mean the startup of a stairway controller.

Reprogramming the device with ETS generally overwrites all parameter values. If scenes are learnt via the bus, overwriting can be suppressed.
Using a common reset object, changed scenes can be reset to the values originally set. For this, a reset is only triggered with a "1" telegram.

- Scene function *(visible if Scene function is activated)*	
Scene A	**deactivated**
	with scene storage function
	without scene storage function
Scene number for Scene A *(visible if Scene A is activated)*	1 – 64 **[1]**
Value of input event for Scene A *(visible if Scene A is activated)*	“0”
	“1”
…	…
Scene H	**deactivated**
	with scene storage function
	without scene storage function
Scene number for Scene H *(visible if Scene H is activated)*	1 – 64 **[8]**
Value of input event for Scene H *(visible if Scene H is activated)*	“0”
	“1”
Stored scenes by ETS download or object Reset *(Note: Reset function/object must be activated in the base settings.)*	**overwriteable**
	not overwriteable
The end of a learning event is signalled by an operating cycle (off/on)	**activated**
	deactivated

No.	Name	Function	C	R	W	T	U
1	R1: Input (DPT 18.001)	Scene	C	–	W	–	–

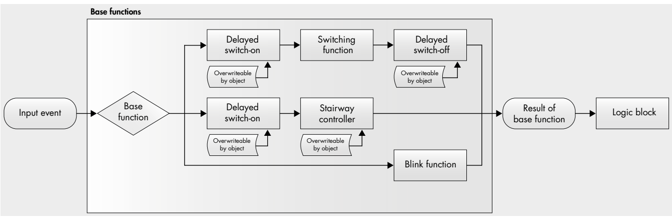
Notes:

- Scene objects had the same priority as input objects, i.e. the last telegram always prevails.
- The defined input result is not passed directly to the relay. Only the Base function and the result from the logic block produce the switching state.
- If scenes are to be reset with a reset object, this function/object must be enabled in the base settings. A reset applies to all selected parameters (not just scenes).

4. Base functions

The basic channel functions are defined in the Base functions. The Switching function, the Stairway controller and the Blink function are actuated by the switching or scene object. It can be defined whether this takes place with a delay. A delay is not possible for the Blink function.

Downstream logic functions are implemented, as opposed to the Base function.



R1: Input events / Filters *(visible if Channel 1 is activated)*	
Base function	Switching
	Stairway controller
	Blink function
Delayed switch-on (switching, scenes, central function) *(visible for switching and stairway control)*	deactivated
	activated
Delayed switch-off (switching, scenes, central function) *(visible for switching)*	deactivated
	activated

4.1 Delayed switch-on and switch-off

For the switching Base function, both a delayed switch-on and a delayed switch-off can be set in the parameters. For the stairway control Base function, a delayed switch-on can be set in the parameters. The delays mean that the Base functions are carried out with a delay, e.g. the channel switches on or the stairway control starts only after the delayed switch-on completes.

Notes:

- If the hours, minutes and seconds parameters for the delays are all set to “0”, no delay time will start, and the channel will switch immediately.
- The common “Parameter Reset” object cancels running timers.

Whether a delay should be started can be defined individually for each object (switching, central and scene objects). So for example, the switching and central objects can act with a delay, but scenes are switched immediately.

Delay times can be made retriggerable, i.e. after receiving the same telegram value again, the time is restarted.

In service, the delay time can be changed via the KNX bus between 0 and 65535 seconds (corresponds to max. 18.2 hours). For this, a telegram with a 2-byte value (0 to 65535) must be sent. Once such a value is received, the duration previously set in the parameters loses its validity (also after a bus reset).

Reprogramming the device with ETS generally overwrites all parameter values. If a delay time is changed/set via the bus, overwriting can be suppressed. Using a common reset object (parameter reset), changed delay times can be reset to the values originally set in the parameters.

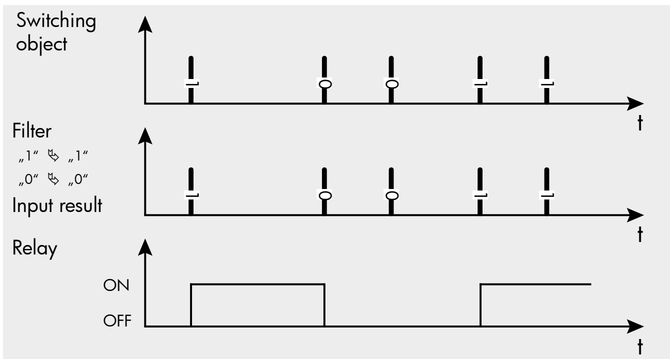


Figure 1: Switching function

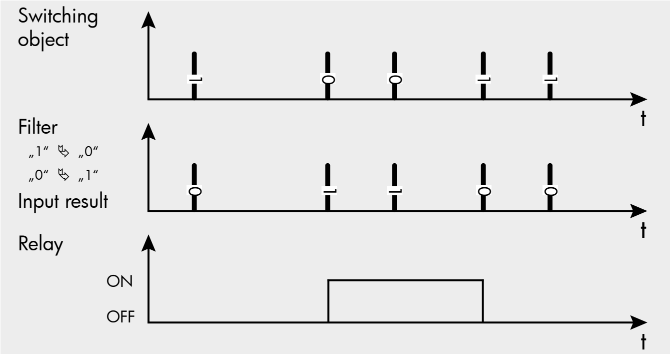


Figure 2: Switching function with filter

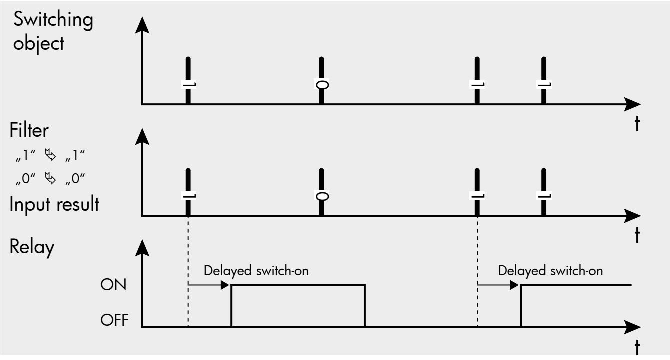


Figure 3: Delayed switch-on

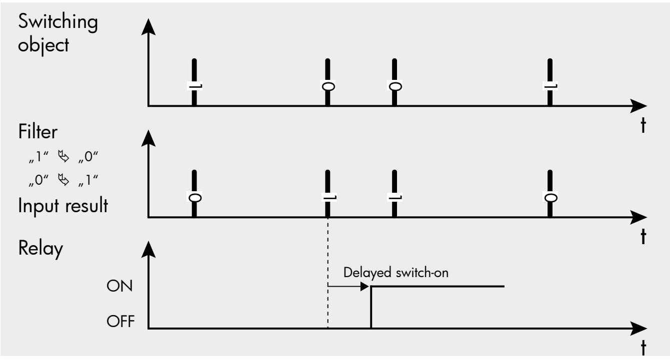


Figure 4: Delayed switch-on with filter

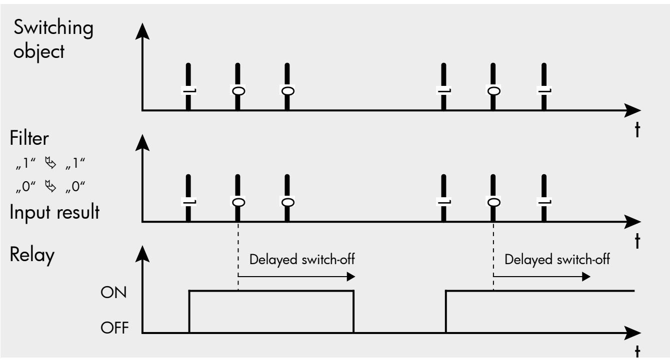


Figure 5: Delayed switch-off, non-retriggerable

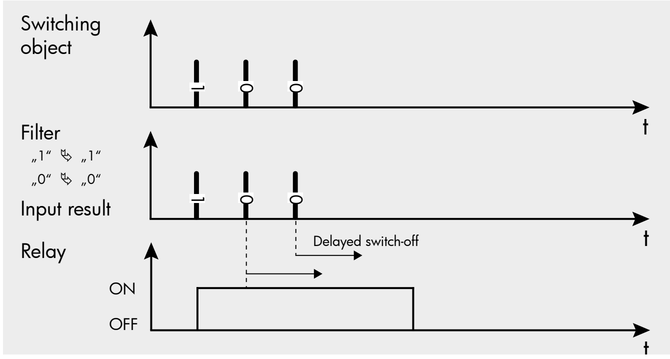


Figure 6: Delayed switch-off, retriggerable

- Delayed switch-on *(visible if Delayed switch-on is activated)*	
Start/retriggering of switch-on delay by	input event “1”
Hours	0-24 **[0]**
Minutes	0-59 **[1]**
Seconds	0-59 **[0]**
Delayed switch-on	**non-retriggerable**
	retriggerable
Switching object works	without delay
	with delay
Central object works	**without delay**
	with delay
Scene object works	**without delay**
	with delay
Delay time	**determined by parameters**
	overwriteable by object
Times changed by object by ETS download or object Reset	**overwriteable**
(visible if “overwriteable by object” is activated) *(Note: Reset function/object must be activated in the base settings.)*	not overwriteable

- Delayed switch-off *(visible if Delayed switch-off is activated)*	
Start/retriggering of switch-off delay by	input event “0”
Hours	0-24 **[0]**
Minutes	0-59 **[1]**
Seconds	0-59 **[0]**
Delayed switch-off	**non-retriggerable**
	retriggerable
Switching object works	**without delay**
	with delay
Central object works	**without delay**
	with delay
Scene object works	**without delay**
	with delay
Delay time	**determined by parameters**
	overwriteable by object

- **Delayed switch-off** *(visible if Delayed switch-off is activated)*	
Times changed by object by ETS download or object Reset *(visible if "overwriteable by object" is activated)* *(Note: Reset function/object must be activated in the base settings.)*	**overwriteable**
	not overwriteable

No.	Name	Function	C	R	W	T	U
0	R1: Input (DPT 1.001)	Switching	C	–	W	–	–
1	R1: Input (DPT 18.001)	Scene	C	–	W	–	–
7	R1: Input (DPT 7.005)	Delayed switch-on time	C	–	W	–	–
8	R1: Input (DPT 7.005)	Delayed switch-off time	C	–	W	–	–
240	General: Input (1.001)	Central switching	C	–	W	–	–

Note:

- If the delay times are to be reset with a reset object, this function/ object must be enabled in the base settings. A reset applies to all selected parameters (not just delay times).

4.2 Stairway controller

If no delay, logical connection or logic functions are switched on, the channel is switched on with the stairway control function by an input event. After expiry of a freely-selectable time (follow-up time), the channel switches off independently. The input event results from the input filter and the input objects.

Notes:

- If the hours, minutes and seconds parameters for the delays are all set to "0", the stairway controller will not start.
- The common "Parameter Reset" object cancels running timers.

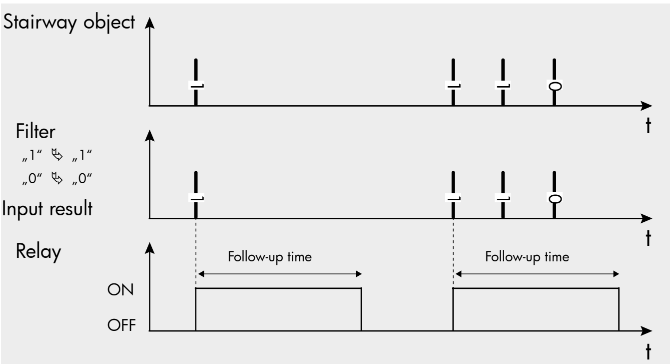


Figure 7: Stairway controller, non-retriggerable

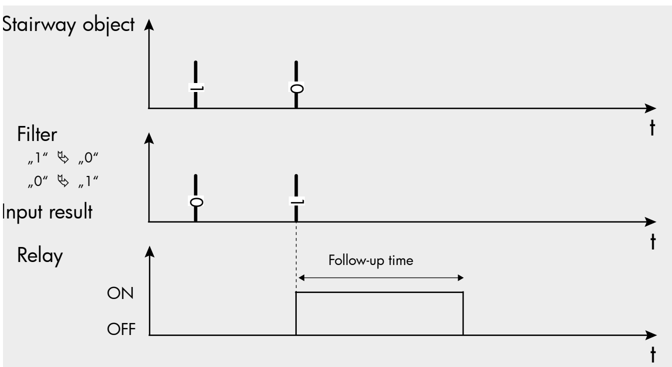


Figure 8: Stairway controller with filter function

The "Stairway controller" Base function can be changed by the setting of parameters. So it can be defined whether the time should be started with a switch-on or switch-off telegram (switching, scene or central object). The follow-up time can for example be set to be retriggerable or not, or can be extended incrementally.

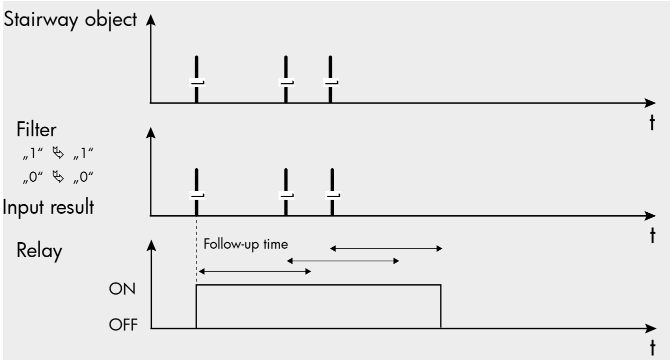


Figure 9: Stairway controller, retriggerable

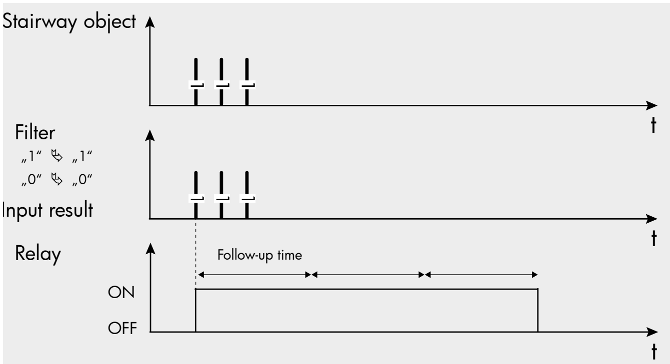


Figure 10: Stairway controller, incremental

Using a delayed switch-on, the follow-up time can be started with a delay. Delayed switch-off is not available for the stairway control function.

The follow-up time can be switched off manually with an off telegram (manual off).

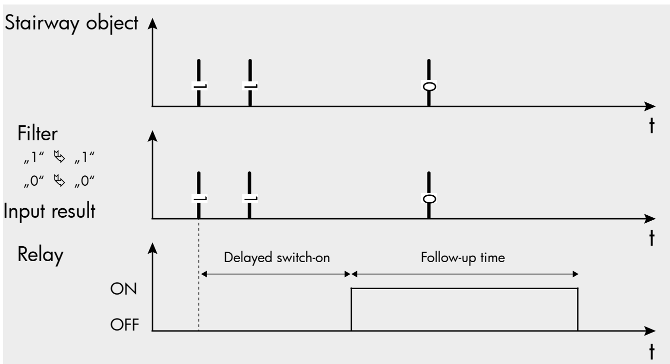


Figure 11: Stairway controller without manual off function and with non-retriggerable delayed switch-on

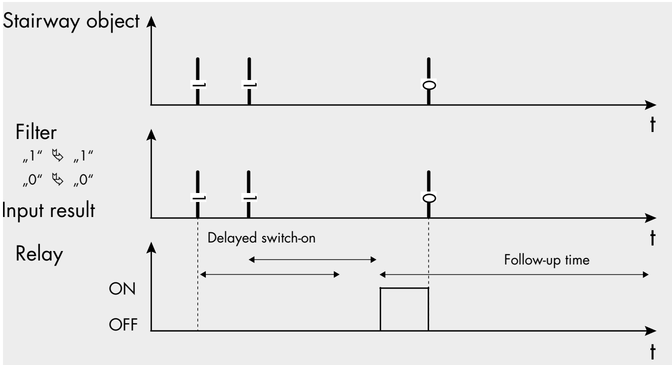


Figure 12: Stairway controller with manual off function and with retriggerable delayed switch-on

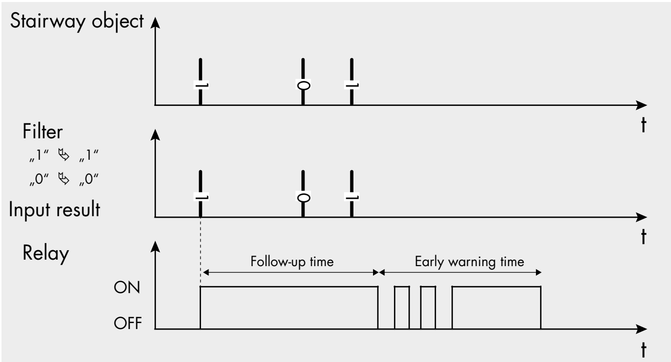


Figure 13: Stairway controller, non-retriggerable without manual off with two early warnings

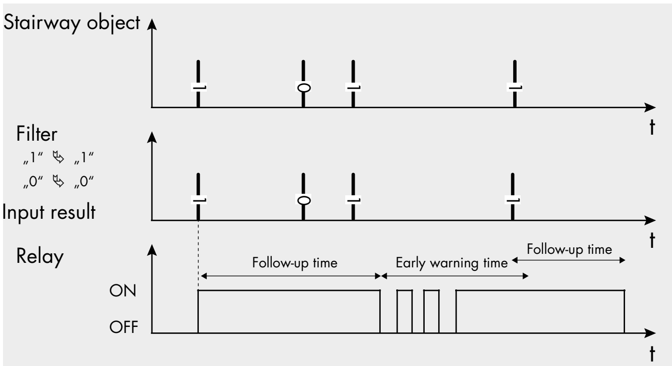


Figure 14: Restart of follow-up time during early warning

In service, the follow-up time can be changed via the KNX bus between 0 and 65535 seconds (corresponds to max. 18.2 hours). For this, a telegram with a 2-byte value (0 to 65535) must be sent. Once such a value is received, the duration previously set in the parameters loses its validity (also after a bus reset).

Reprogramming the device with ETS generally overwrites all parameter values. If a follow-up time is changed/set via the bus, overwriting can be suppressed. Using a common reset object, a changed follow-up time can be reset to the values originally set.

- Stairway controller *(visible if Stairway controller is activated)*	
Start/retriggering of stairway controller by	Input event "1"
Manual off of stairway controller by	Input event "0"
Hours	0-24 **[0]**
Minutes	0-59 **[5]**
Seconds	0-59 **[0]**
Stairway controller	**without manual off**
	with manual off

- Stairway controller *(visible if Stairway controller is activated)*	
Follow-up time	**non-retriggerable**
	retriggerable
	retriggerable, incremental
Maximum increments *(visible on "retriggerable, incremental")*	2 - 5 **[3]**
Early warning	**deactivated**
	activated
Early warning time in seconds *(visible if Early warning is activated)*	5 – 255 **[30]**
Number of early warnings at start of early warning time *(visible if Early warning is activated)*	1 – 3 **[3]**
Follow-up time	**determined by parameters**
	overwriteable by object
Times changed by object by new ETS download or object Reset *(visible if "overwriteable by object")* *(Note: Reset function/object must be activated in the basic settings.)*	**overwriteable**
	not overwriteable

No.	Name	Function	C	R	W	T	U
0	R1: Input (DPT 1.001)	Switching	C	–	W	–	–
1	R1: Input (DPT 18.001)	Scene	C	–	W	–	–
8	R1: Input (DPT 7.005)	Follow-up time	C	–	W	–	–
240	General: Input (1.001)	Central switching	C	–	W	–	–

Note:

- If the follow-up time is to be reset with a reset object, this function/object must be enabled in the base settings. A reset applies to all selected parameters (not just the follow-up time).

4.3 Blink function

With the Blink function, the channel periodically switches on and off, for example in order to make an LED blink in a caretaker's office, indicating that a certain door has been opened. If the input event is "1", the Blink function is started, and if it is "0", it is stopped (switching, central and scene object).

The Blink function cannot be switched on or off with a delay.

The response object shows whether the Blink function is switched on or off, and not whether the relay is closed or open. To keep the bus loading low, in this case the current relay value is not sent to the bus.

On and off times can be set from 1 to 59 seconds. To protect the relay from greater loads, it is not possible to set times under 1 second. The lowest possible frequency is 0.5 Hz (1 second on and 1 second off).

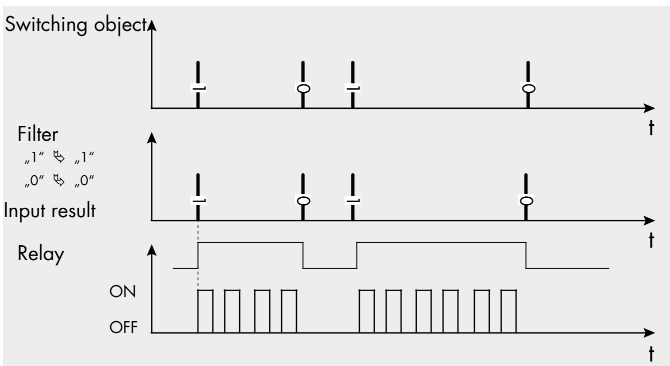


Figure 15: Symmetrical Blink function

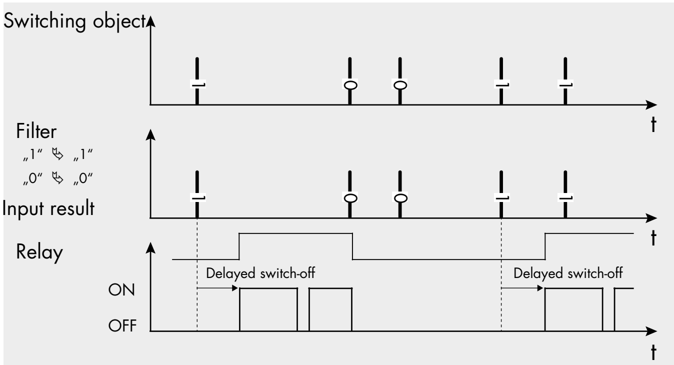


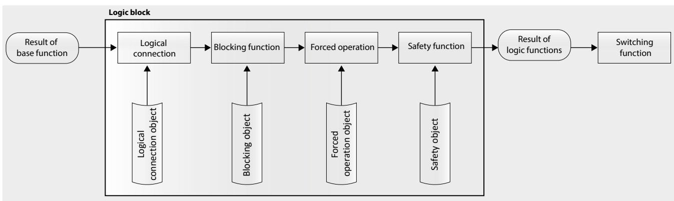
Figure 16: Asymmetrical Blink function

- Blink function *(visible if Blink function is activated)*	
Blink with	Input event “1”
Off with	Input event “0”
On time in seconds	1-59 **[4]**
Off time in seconds	1-59 **[4]**

No.	Name	Function	C	R	W	T	U
0	R1: Input (DPT 1.001)	Switching	C	–	W	–	–
1	R1: Input (DPT 18.001)	Scene	C	–	W	–	–
240	General: Input (1.001)	Central switching	C	–	W	–	–

5. Logic functions

The actuator has four logic functions: Logical connection, Blocking, Forced operation and Safety. Their sequencing gives their priority, i.e. the Safety function has the highest priority, as it is at the end of the chain. If the Safety function is activated by the Safety object, the results from the Base function, Logical connection, Blocking function and Forced operation blocks are not sent to the switch output.



Effects of Logic functions on Base functions:

The Blocking, Forced operation and Safety functions affect the Base functions. Once one of these Logic functions is activated, any channel times running are immediately ended. The result of the Base function is that which would have occurred if the time had run out normally (setting: follows lower level state).

Examples:

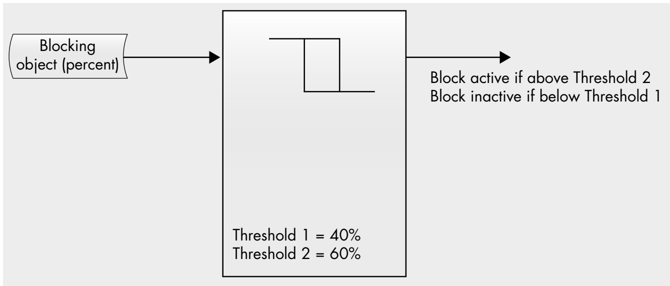
1. The Blocking function is activated during a delayed switch-on. The Base function supplies the result “1” in the background and the delay time is ended immediately.
2. If forced operation is activated during a delayed switch-off, the time is also ended, but the result of the Base function is “0”.
3. For follow-up times that are running, the result is “0”, as a stairway controller automatically switches off. The result is also “0” if a delayed switch-on has been previously selected, independently of whether the Logic function was activated during the delayed switch-on or during the follow-up time.

Only Logical connection functions allow times that are running to continue running in the background. Therefore the result of the Base function depends on the point in time at which the Logical connection was deactivated again.

Inputs for Logic functions / Comparators:

Up to Forced operation, logic functions are controlled by 1-bit objects / values. For example, a channel can be blocked by a blocking object. With a Logical connection, the logical connection object is connected logically / by Boolean logic with the result of the Base function, e.g. with an AND operation.

As an alternative to these 1-bit objects, the functions (except forced operation) can also be activated by a Comparator. Instead of a 1-bit blocking object, an object is now output with another format, e.g. percent, 2-byte integer, floating point etc. For a comparator function, two threshold values can be freely selected. The object values are compared with these two threshold values. If values go above or below these thresholds, the logic function is activated or deactivated. A suitable choice of threshold values enables the creation of a comparator with integrated hysteresis.

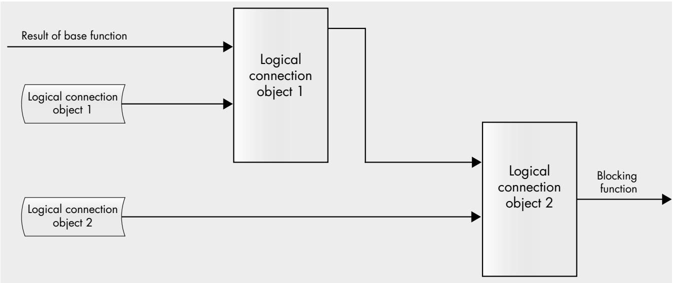


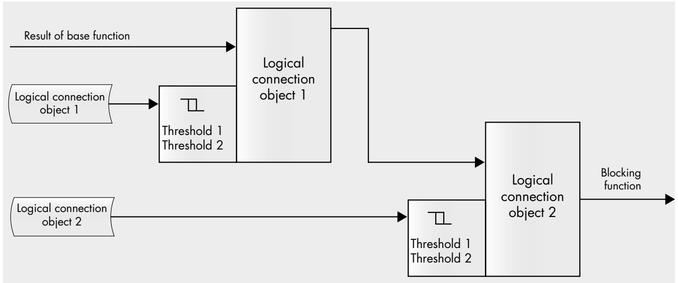
R1: Logic functions *(visible if Channel 1 is activated)*	
Logical connection 1	**deactivated**
	binary / 1 bit
	extended / comparator
Logical connection 2	**deactivated**
	binary / 1 bit
	extended / comparator
Blocking	**deactivated**
	binary / 1 bit
	extended / comparator
Forced operation	**deactivated**
	activated
Safety	**deactivated**
	binary / 1 bit
	extended / comparator

5.1 Logical connection function

For Logical connection functions Boolean algebra is used. The following functions are available: AND, OR and XOR.

There are two sequentially-switched Logical connection functions/ gates available. Logical connection function 1 has Logical connection object 1 and the result of the Base function as input. Logical connection function 2 has Logical connection object 2 and the result of Logical connection function 1 as input. The result of Logical connection 2 is passed on to the next logic function.





It can be set in the parameters whether the logical connection objects should act on the Logical connection function as inverted, and the value that the connection should have after bus voltage recovery can also be set.

Times that are running, e.g. delay times and follow-up times, are not stopped or ended when a logical connection is activated. If for example a logical connection is activated during a follow-up time for a stairway controller, the behaviour of the output when the logical connection is deactivated depends on whether the follow-up time has expired or not.

- Logical connection 1: binary *(visible if "Logical connection x: binary / 1 bit" is activated)*	
The result of the Base function is connected with Logical connection object 1.	
Logical connection 1	**OR**
	AND
	XOR
Evaluation of Logical connection object 1	**normal**
	inverted
Value of Logical connection object 1 after bus voltage recovery	**"0"**
	"1"

- Logical connection 2: binary *(visible if "Logical connection x: binary / 1 bit" is activated)*	
The result of Logical connection 1 is connected with Logical connection object 2.	
Logical connection 2	**OR**
	AND
	XOR
Evaluation of Logical connection object 2	**normal**
	inverted
Value of Logical connection object 2 after bus voltage recovery	**"0"**
	"1"

No.	Name	Function	C	R	W	T	U
2	R1: Input (DPT 1.001)	Logical connection 1	C	–	W	–	–
3	R1: Input (DPT 1.001)	Logical connection 2	C	–	W	–	–

Instead of a 1-bit logical connection object, the result of a Comparator can also apply.

- Logical connection 1: extended *(visible if "Logical connection: extended / comparator" is activated)*	
The results of the Base function and the comparator are connected.	
Logical connection object 1 is compared with threshold values 1 and 2.	
Logical connection 1	**OR**
	AND
	XOR
Comparator format	**1 byte percent (DPT5.001)**
	1 byte counter (DPT5.010)
	1 byte counter with prefix (DPT6.010)
	2 byte float (DPT9.00x)
	2 byte counter (DPT7.00x)
	2 byte counter with prefix (DPT8.00x)
	4 byte float (DPT14.00x)
	4 byte counter (DPT12.00x)
	4 byte counter with prefix (DPT13.00x)
Result of the comparison is "1" if	**Logical connection object 1 >= Threshold value 1**
	Logical connection object 1 <= Threshold value 1
Threshold 1	0 – 100 **[60]**
Result of the comparison is "0" if	Logical connection object 1 >= Threshold value 2
	Logical connection object 1 <= Threshold value 2
Threshold 2	0 - 100 **[40]**
Comparator value after bus voltage recovery	**"0"**
	"1"

- Logical connection 2: extended *(visible if "Logical connection: extended / comparator" is activated)*	
The results from Logical connection 1 and the comparator are connected.	
Logical connection object 2 is compared with threshold values 1 and 2.	
Logical connection 2	**OR**
	AND
	XOR

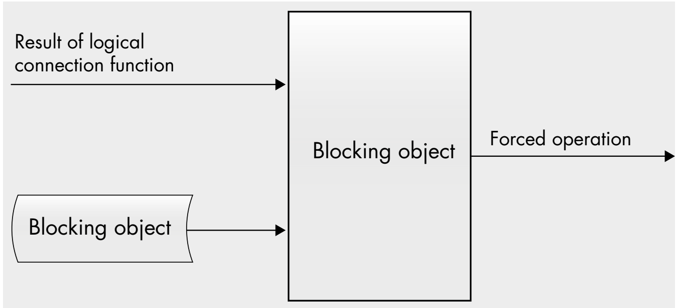
- **Logical connection 2: extended** *(visible if "Logical connection: extended / comparator" is activated)*	
Comparator format	**1 byte percent (DPT5.001)**
	1 byte counter (DPT5.010)
	1 byte counter with prefix (DPT6.010)
	2 byte floating point (DPT9.00x)
	2 byte counter (DPT7.00x)
	2 byte counter with prefix (DPT8.00x)
	4 byte floating point (DPT14.00x)
	4 byte counter (DPT12.00x)
	4 byte counter with prefix (DPT13.00x)
Result of the comparison is "1" if	**Logical connection object 2 >= Threshold value 1**
	Logical connection object 2 <= Threshold value 1
Threshold 1	0 – 100 **[60]**
Result of the comparison is "0" if	Logical connection object 2 >= Threshold value 2
	Logical connection object 2 <= Threshold value 2
Threshold 2	0 - 100 **[40]**
Comparator value after bus voltage recovery	**"0"**
	"1"

No.	Name	Function	C	R	W	T	U
2	R1: Input (DPT5.001)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT5.010)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT6.010)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT9.x)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT7.x)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT8.x)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT14.x)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT12.x)	Logical connection 1	C	–	W	–	–
2	R1: Input (DPT13.x)	Logical connection 1	C	–	W	–	–
3	R1: Input (DPT5.001)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT5.010)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT6.010)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT9.x)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT7.x)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT8.x)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT14.x)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT12.x)	Logical connection 2	C	–	W	–	–
3	R1: Input (DPT13.x)	Logical connection 2	C	–	W	–	–

5.2 Blocking function

The Blocking function is controlled by the blocking object and by the lower level functions. The lower level function is the logical connection function, and if this is not activated, it is the result of the Base function (switching, stairway, blinking). The blocking function is activated by the blocking object. The object value ("1" or "0") with which this should happen can be selected.

The result of the blocking function is passed either to the higher level logic functions (Forced operation, Safety), if these are activated, or to the switch output. The next higher level logic function is Forced operation.



The result of the active blocking function can be selected. It can be "0", "1" or "no reaction". "No reaction" at the beginning of the blocking means that the current result (the one present when the block was activated) is frozen during the block.

The result can also be defined on removal of the block. It can be selected as a specific value, "0" or "1". This value is passed on to the higher level function when the block is removed. If "no reaction" is set, the current result of the blocking remains in force. Only an input event updates the result. If "follows lower level state" is set, the blocking function determines a new result based on the lower level functions.

If the result of the blocking function when the block is removed is "1", on a stairway controller the follow-up time will be started. For stairway controllers with delayed switch-on, the delayed switch-on is ignored.

The blocking function can be time-limited, i.e. an activated block can be automatically deactivated after a chosen period.

The value that the blocking object should have after bus voltage recovery can be set.

Notes:

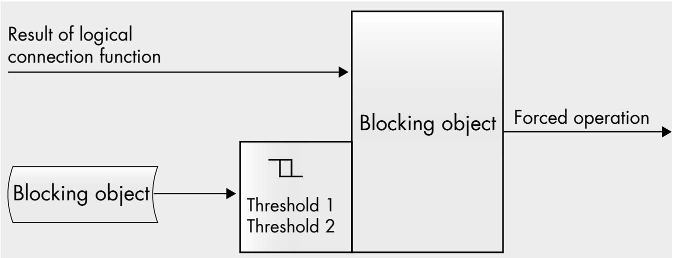
- Activation and deactivation take place without a delay, i.e. the delay times set in the parameters are ignored.
- If the blocking function is activated, any running delay times and follow-up times from Base functions are cleared.
- If the hours, minutes and seconds parameters for the time limit are all set to "0", the time limit will not be activated.

- **Blocking: binary** *(visible if "Blocking function: binary / 1 bit" is activated)*	
The result of Logical connection 2 is passed on depending on the Blocking object.	
Blocking active on blocking object value	"0"
	"1"
Action on start of block	"0"
	"1"
	no reaction
Value of Base function when block removed (no priority active)	"0"
	"1"
	no reaction
	follows lower level state
Blocking function time-limited	**deactivated**
	activated

- Blocking: binary *(visible if "Blocking function: binary / 1 bit" is activated)*	
Hours *(visible if activated)*	0-24 **[0]**
Minutes *(visible if activated)*	0-59 **[10]**
Seconds *(visible if activated)*	0-59 **[0]**
On bus voltage recovery	**not blocked**
	blocked

No.	Name	Function	C	R	W	T	U
4	R1: Input (DPT 1.001)	Blocking	C	–	W	–	–

Instead of a 1-bit blocking object, the result of a Comparator can also apply.



- Blocking function: extended *(visible if "Blocking function: extended / comparator" is activated)*	
The result of Logical connection 2 is passed on depending on the Blocking object.	
The Blocking object is compared with threshold values 1 and 2.	
Comparator format	**1 byte percent (DPT5.001)**
	1 byte counter (DPT5.010)
	1 byte counter with prefix (DPT6.010)
	2 byte floating point (DPT9.x)
	2 byte counter (DPT7.x)
	2 byte counter with prefix (DPT8.x)
	4 byte floating point (DPT14.x)
	4 byte counter (DPT12.x)
	4 byte counter with prefix (DPT13.x)
Block is active if	**Blocking object >= Threshold value 1**
	Blocking object <= Threshold value 1
Threshold 1	0 – 100 **[60]**
Block is inactive if	**Blocking object >= Threshold value 2**
	Blocking object <= Threshold value 2
Threshold 2	0 – 100 **[40]**
Action on start of block	"0"
	"1"
	no reaction

- Blocking function: extended *(visible if "Blocking function: extended / comparator" is activated)*	
Value of Base function when block removed (no priority active)	"0"
	"1"
	no reaction
	follows lower level state
Blocking function time-limited	**deactivated**
	activated
Hours *(visible if activated)*	0-24 **[0]**
Minutes *(visible if activated)*	0-59 **[10]**
Seconds *(visible if activated)*	0-59 **[0]**
On bus voltage recovery	**not blocked**
	blocked

No.	Name	Function	C	R	W	T	U
4	R1: Input (DPT5.001)	Blocking	C	–	W	–	–
4	R1: Input (DPT5.010)	Blocking	C	–	W	–	–
4	R1: Input (DPT6.010)	Blocking	C	–	W	–	–
4	R1: Input (DPT9.x)	Blocking	C	–	W	–	–
4	R1: Input (DPT7.x)	Blocking	C	–	W	–	–
4	R1: Input (DPT8.x)	Blocking	C	–	W	–	–
4	R1: Input (DPT14.x)	Blocking	C	–	W	–	–
4	R1: Input (DPT12.x)	Blocking	C	–	W	–	–
4	R1: Input (DPT13.x)	Blocking	C	–	W	–	–

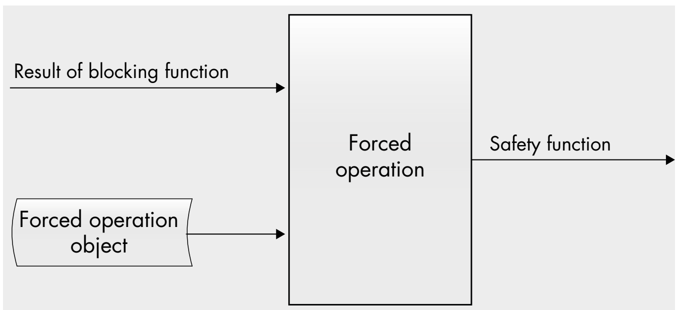
5.3 Forced operation

Forced operation is activated and deactivated by 2-bit Forced operation objects. The channel is switched to high-priority switching status using 2-bit telegrams. If the 2-bit Forced operation object receives a telegram whose first bit (Bit 1) has the value "1", then Forced operation is active. In this case, the switch actuator switches to the state defined by the second bit (Bit 0) of the telegram.

Bit 1	Bit 0	Function
1	1	Forced operation active "1"
1	0	Forced operation active "0"
0	1	Forced operation inactive
0	0	Forced operation inactive

Forced operation is controlled by the Forced operation object and by the lower level functions. Lower level functions are the Blocking and Logical connection functions, and if these are not activated, the result of the Base function (switching, stairway, blinking). Forced operation is activated by the forced operation object.

The result of Forced operation is passed either to the higher level logic function (Safety) if this is activated, or to the switch output.



The result can also be defined on removal of Forced operation. It can be selected as a specific value, “0” or “1”. This value is passed on to the higher level function, Safety, when the function is removed. If “no reaction” is set, the current result of Forced operation remains in force. Only an input event updates the result. If “follows lower level state” is set, Forced operation determines a new result based on the lower level functions.

If the result when Forced operation is removed is “1”, then for a stairway controller, the follow-up time is started. For stairway controllers with delayed switch-on, the delayed switch-on is ignored.

The value that the Forced operation object should have after bus voltage recovery can be set.

Notes:

- Activation and deactivation take place without a delay, i.e. the delay times set in the parameters are ignored.
- If Forced operation is activated, any running delay times and follow-up times from Base functions are cleared.

- Forced operation *(visible if Forced operation is activated)*	
The result of the Blocking function is passed on depending on the Forced operation.	
Value of Base function when forced operation removed (no priority active)	“0”
	“1”
	no reaction
	follows lower level state
On bus voltage recovery	forced operation = OFF
	forced operation = ON
	no forced operation

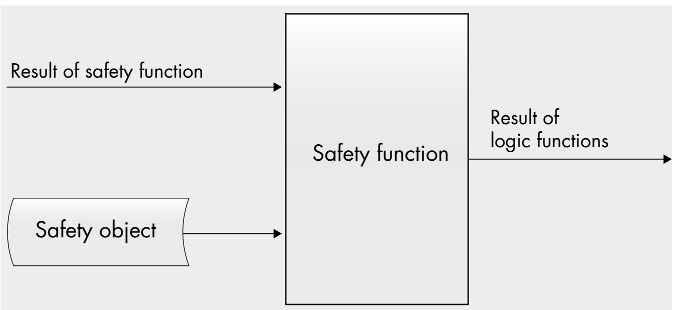
No.	Name	Function	C	R	W	T	U
5	R1: Input (DPT 2.001)	Forced operation	C	–	W	–	–

5.4 Safety function

The Safety function has the highest priority in the sequence of functions. The Safety function is an extended Blocking function. Here, the Safety object is additionally monitored, i.e. telegrams must be received to this object periodically. Otherwise the channel goes into the subsequent state defined in the parameters.

The Safety function is controlled by the Safety object and by the previous functions. Previous functions are the Logical connection, Blocking and Forced operation functions, and if these are not activated, the result of the Base function (switching, stairway, blinking). The Safety function is activated by the Safety object. The object value (“1” or “0”) with which this should happen can be selected. If the object value is absent in a defined timeframe, the Safety function is also activated.

The result of the Safety function is passed to the switch output.



The result of the active Safety function can be selected. It can be “0”, “1” or “no reaction”. “No reaction” at the beginning of the Safety function means that the current result (the one present when the Safety function was activated) is frozen during the block.

The result can also be defined on deactivation of the Safety function. It can be selected as a specific value, “0” or “1”. This value is passed on to the switching channel when the function is deactivated. If “no reaction” is set, the current result remains in force. Only an input event updates the result. If “follows lower level state” is set, the Safety function determines a new result based on the lower level functions.

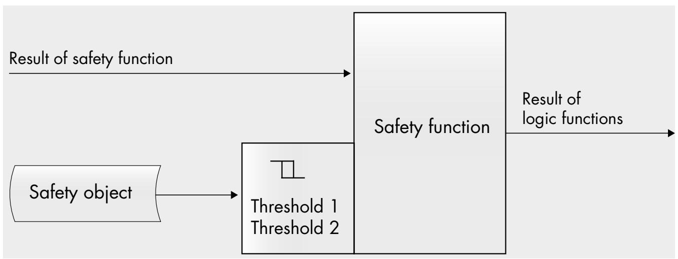
If the result of the Safety function when removed is “1”, then for a stairway controller, the follow-up time is started. For stairway controllers with delayed switch-on, the delayed switch-on is ignored.

The value that the Safety object should have after bus voltage recovery can be set.

- Safety: binary *(visible if “Safety function: binary / 1 bit” is activated)*	
The result of Forced operation is passed on depending on the Safety object.	
Safety function active with Safety object value	“0”
	“1”
Action on start of Safety function	“0”
	“1”
	no reaction
Value of Base function when function removed (no priority active)	“0”
	“1”
	no reaction
	follows lower level state
Periodic monitoring	**deactivated**
	activated
Minutes (visible if activated)	1 - 255 **[10]**
On bus voltage recovery	**not blocked**
	blocked

No.	Name	Function	C	R	W	T	U
6	R1: Input (DPT 1.001)	Safety	C	–	W	–	–

Instead of a 1-bit Safety object, the result of a Comparator can also apply.

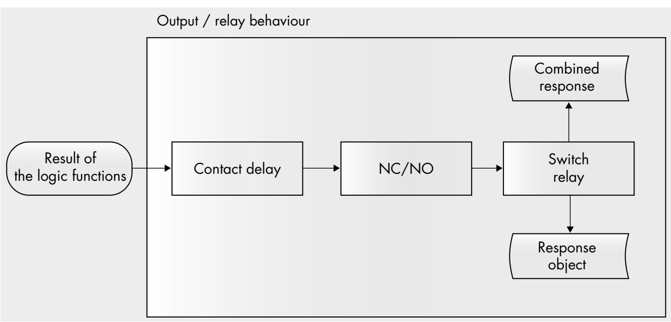


- Safety: extended *(visible if “Safety function: extended / comparator” is activated)*	
The result of Forced operation is passed on depending on the Safety object.	
The Safety object is compared with threshold values 1 and 2.	

- Safety: extended *(visible if "Safety function: extended / comparator" is activated)*	
Comparator format	**1 byte percent (DPT5.001)**
	1 byte counter (DPT5.010)
	1 byte counter with prefix (DPT6.010)
	2 byte floating point (DPT9.x)
	2 byte counter (DPT7.x)
	2 byte counter with prefix (DPT8.x)
	4 byte floating point (DPT14.x)
	4 byte counter (DPT12.x)
	4 byte counter with prefix (DPT13.x)
Safety function is active if	**Blocking object >= Threshold 1**
	Blocking object <= Threshold value 1
Threshold 1	0 – 100 **[60]**
Safety function is inactive if	Blocking object >= Threshold value 2
	Blocking object <= Threshold value 2
Threshold 2	0 – 100 **[40%]**
Action on start of Safety function	"0"
	"1"
	no reaction
Value of Base function when function removed (no priority active)	"0"
	"1"
	no reaction
	follows lower level state
Periodic monitoring	**deactivated**
	activated
Minutes *(visible if activated)*	1 - 255 **[10]**
On bus voltage recovery	**not blocked**
	blocked

No.	Name	Function	C	R	W	T	U
6	R1: Input (DPT5.001)	Safety	C	–	W	–	–
6	R1: Input (DPT5.010)	Safety	C	–	W	–	–
6	R1: Input (DPT6.010)	Safety	C	–	W	–	–
6	R1: Input (DPT9.x)	Safety	C	–	W	–	–
6	R1: Input (DPT7.x)	Safety	C	–	W	–	–
6	R1: Input (DPT8.x)	Safety	C	–	W	–	–
6	R1: Input (DPT14.x)	Safety	C	–	W	–	–
6	R1: Input (DPT12.x)	Safety	C	–	W	–	–
6	R1: Input (DPT13.x)	Safety	C	–	W	–	–

6. Output / relay behaviour



6.1 Contact delay function

The Contact delay function is for protection against overloads on the power network. Using a central object, channels can be switched simultaneously. Simultaneous switching of many connected loads can lead to short-term overloading of the power network. This problem can be evened out using the Contact delay function. Thus the switch command is only given to the relay after a delay time has expired. These delay times are not to be confused with delayed switch-on and delayed switch-off. They are much shorter.

The Contact delay function also makes it possible to prioritise switching channels. If for example all channels are to be switched on by a central command, the shortest contact delay time is given to the channel that should be switched first.

R1: Output / relay behaviour *(visible if Channel 1 is activated)*	
Contact delay function	**deactivated**
	activated
Contact delay when switching on in milliseconds *(visible if activated)*	10 – 10000 **[100]**
Contact delay when switching off in milliseconds *(visible if activated)*	10 – 10000 **[100]**

6.2 Contact type

The switch actuator is fitted with latching relays. Under the Contact type parameter, each channel can be defined to have relays behave as NO (normally open) or NC (normally closed). For NC operation, the value defined by the Base function and logic functions is inverted.

R1: Output / relay behaviour *(visible if Channel 1 is activated)*	
Contact type	**NO**
	NC

6.3 Responses

The actuator switches the switch contact as soon as the base and logic functions give a switch command. After that, a Response object is also generated, i.e. no true measurement takes place of whether a relay has actually switched. Relay or load defects do not therefore appear. For actuators with current detection, the response value can also be taken via current/output detection. Here, the actuator measures whether current is actually flowing.

R1: Output / relay behaviour *(visible if Channel 1 is activated)*	
Responses	**deactivated**
	Detection via relay state
	Detection via value of current
	Detection via active power

For each channel, it is possible for the response to be via a 1-bit object and/or a 32-bit combined response (KNX DPT27.001). For a combined response, Bit 0 corresponds to the state of Channel R1 and Bit 3 the state of Channel K4. The option can be selected whether a channel should be included in the combined response.

Note:

- DPT27.001 describes a 32-bit long object. The first two bytes show the state, the last two bytes the validity. For the response "Channel R1 closed", Bit 0 and Bit 16 are set. If Channel R1 is not to be included in the combined response (parameter), Bit 16 is cleared.

- Responses via relay / current / active power *(visible if Responses is activated)*	
Channel with combined response, 16-bit	not taken into account
	taken into account

Here too, the type of response can be defined (active response object / passive status object).

- Responses via relay / current / active power *(visible if Responses is activated)*	
Response, 1-bit	deactivated
	active response object
	passive status object

The response value follows the state of the switch contact. Either the original state (closed = "1" / open = "0") or the inverted value (closed = "0" / open = "1") can be sent. This applies both to individual and to combined responses.

- Responses via relay *(visible on detection via relay state)*	
Response when relay open	**"0"**
	"1"
	no reaction
Response when relay closed	"0"
	"1"
	no reaction

No.	Name	Function	C	R	W	T	U
9	R1: Output (DPT 1.001)	Response	C	–	–	T	–
9	R1: Output (DPT 1.001)	Status object	C	R	–	–	–
242	General output (DPT 27.001)	Combined response (active)	C	–	–	T	–
242	General output (DPT 27.001)	Combined response (passive)	C	R	–	–	–

6.3.1 Detection via value of current or active power

The response can be derived via current detection or active power measurement.

For this, upper and lower current thresholds must be set. These thresholds set out where the channel should be regarded as closed or open. The values for the response are determined on this basis.

When a contact is being closed, there are usually pulses of interference, which can be caused by the relay bouncing and by connected loads (inductive, capacitive, etc.). In order that no erroneous responses are sent to the KNX bus during switch-on, current measurement can be started with a delay.

- Responses via current *(visible on detection via value of current)*	
Delay in evaluation after relay closing in seconds	0 – 60 **[10]**

- Responses via current *(visible on detection via value of current)*	
Lower threshold in milliamps	0 – 16000 **[8000]**
Response when values reach or go below lower threshold value	**"0"**
	"1"
	no reaction
Upper threshold in milliamps	0 – 16000 **[12000]**
Response when values reach or go over upper threshold value	"0"
	"1"
	no reaction

- Responses via active power *(visible on detection via active power)*	
Delay in evaluation after relay closing in seconds	0 – 60 **[10]**
Lower threshold in Watts	0 – 16000 **[1600]**
Response when values reach or go below lower threshold value	**"0"**
	"1"
	no reaction
Upper threshold in Watts	0 – 16000 **[2400]**
Response when values reach or go over upper threshold value	"0"
	"1"
	no reaction

No.	Name	Function	C	R	W	T	U
9	R1: Output (DPT 1.001)	Response	C	–	–	T	–
9	R1: Output (DPT 1.001)	Status object	C	R	–	–	–

7. Bus voltage loss and recovery behaviour

If bus voltage is lost, it is possible to switch a switch contact to a last defined position (closed, open).

Note:

- The Contact type parameter (NC / NO) is not taken into account here.

When there is a bus voltage loss event, the actuator stores internally the last valid result of the Base function, although times that are running are disregarded. These stored values can be reactivated on bus voltage recovery.

Behaviour on bus voltage recovery is also selectable. Any parameter values do not directly affect the relay, but affect the result of the Base function. Background for this are the logic functions. It is also possible to define bus voltage recovery behaviour for logic functions. This is of higher level than the Base functions. Only if no logic functions have been set in the parameters do the bus voltage recovery behaviour parameters directly affect the switching channel.

R1: General settings *(visible if Channel 1 is activated)*	
Relay state after bus voltage loss	open
	closed
	no change

R1: General settings *(visible if Channel 1 is activated)*	
Result of Base function after bus voltage recovery	**"0"**
	"1"
	no change
	same as before bus voltage loss

Part C - Measurement of current

8. Measurement methods (Parameters: General Current / Energy)

The switch actuator offers the option of measuring current / calculating energy. Each channel has its own current sensor. There are two different measurement methods available for measuring current / calculating energy. The selected measurement method is applied to all channels.

8.1 Method 1: Measurement with neutral conductor connected

When the neutral conductor is connected, the phase state of Channel 1 can automatically be determined. The mains frequency in this setting is automatically determined in any case.

The phase of Channel 1 is defined as L1. For the other channels the three different phases can be connected as required (clockwise phase shift L1, L2, L3). Using parameters, the software must now be told which phase is where.

For measurements, the voltage values (effective values) of the individual phases must be known.

The actuator includes one current sensor per channel. For all channels together, it has zero cross detection for the mains voltage.

Current is measured several times over a period and made available as an effective value.

Power is the product of current and voltage. The current is measured several times during a period. The actuator presupposes a sine wave on the voltage. The effective value of the voltage is given in the parameters. The time reference between current measurement and corresponding voltage value is produced with the aid of zero cross switching. The power is also an averaged value.

The energy value is now determined based on the measured power and a time interval.

As the voltage curve is regarded as a sine wave and not measured, the values measured do not correspond exactly to the active power/ energy. The closer the voltage is to a sine wave, the more accurate is the result.

General Current / Energy	
Automatically determine phase state (Neutral conductor required)	deactivated
	activated
L1 voltage value for energy calculation in V	100 – 277 **[230]**
L2 voltage value for energy calculation in V	100 – 277 **[230]**
L3 voltage value for energy calculation in V	100 – 277 **[230]**

General Current / Energy	
Clockwise phase shift L1, L2, L3 Channel 1 (reference to phase state)	L1
Channel 2	**L1**
	L2
	L3
...	...
Channel 8	**L1**
	L2
	L3
Number of data packets for calculation of average	3 – 50 **[10]**

Note:

- In order to measure the current sufficiently accurately, it is read several times in a row and given as an average. The parameter "Number of data packets for calculation of average" affects this. Small values result in quick measurement, but this can be less accurate. Using this parameter, interference effects in the system (voltage fluctuation, spikes in current) can be averaged out.

8.2 Method 2: Measurement without neutral conductor connected

If no neutral conductor is connected to the actuator, the phase state and mains frequency cannot be automatically determined. For each channel an individual cos φ must now be determined. By contrast, the (average) mains voltage and mains frequency are given for all channels.

Power is calculated as follows: V x I x cos φ. Here, V and cos φ are the values given in the parameters, and I the current measured by the actuator on the relevant channel.

This method is much less accurate than the method with a neutral conductor connected. Generally, the phase angle is not known, can only be measured with difficulty, or may change in service, as different loads are connected.

Current measurement / Energy calculation	
Automatically determine phase state (Neutral conductor required)	**deactivated**
	activated
Voltage value for energy calculation in V	100 – 277 **[230]**
Mains frequency	**50 Hz**
	60 Hz
Channel 1 cos φ 0, 100 corresponds to cos φ = 1	1 – 100 **[100]**
...	...
Channel 8 cos φ 0, 100 corresponds to cos φ = 1	1 – 100 **[100]**
Number of data packets for calculation of average	3 – 50 **[10]**

Note:

- In order to measure the current sufficiently accurately, it is read several times in a row and given as an average. The parameter "Number of data packets for calculation of average" affects this. Small values result in quick measurement, but this can be less accurate. Using this parameter, interference effects in the system (voltage fluctuation, spikes in current) can be averaged out.

9. Current measurement and derived functions

Each switching channel on the actuator includes its own current sensor. The current sensor measures the current that flows when the switch contact is closed. The current measured by the switch actuator (generally) serves as the basis for the Current measurement / Energy calculation, Current monitoring, Service hours counter and Operating cycle counter functions.

R1: General settings *(visible if Channel 1 is activated)*	
Current measurement / Energy calculation	**deactivated**
	activated
Current monitoring	**deactivated**
	fixed
	adaptive
Service hours counter	**deactivated**
	activated
Operating cycle counter	**deactivated**
	activated

Note:

- When an output contact is being switched (on), there are usually pulses of interference, which can be caused by the relay bouncing and by connected loads (inductive, capacitive, etc.). So that no false states or values are sent to the KNX bus during switching, current measurement can be started with a time delay after the switching event. During this delay, the value of the current sent is 0 A.

9.1 Current measurement / Energy calculation

The current measured can be sent to the bus periodically. As well as periodic reporting, the value can also be sent when there are large changes. The size of the change required can be selected. The current is sent to the bus in mA.

When an output contact is being switched (on), there are usually pulses of interference, which can be caused by the relay bouncing and by connected loads (inductive, capacitive, etc.). So that no false states or values are sent to the KNX bus during switching, current measurement can be started with a time delay after the switching event. During this delay, the value of the current sent is 0 A.

Measurement of current value/active power as well as the energy value can be carried out for the channel. The measurement methods are set out on the general sheet called "General Current / Energy" (see also section 8). The power is made available in W (Watts) and the energy in kWh.

The currently-measured energy value can be deliberately cleared by the "Reset energy value" object. If all energy values (multiple channels) are to be reset, this can take place with the "Parameter Reset" object. The measured value can also be reset by a new ETS download. In any case, this can be prevented by a parameter.

Note:

- If the hours, minutes and seconds parameters for the cycle times are all set to "0", a cycle time of 1 second will be used.

R1: Current / energy *(visible if current measurement / energy calculation is activated)*	
Delay in evaluation after relay closing in seconds	0 – 60 **[10]**

R1: Current / energy *(visible if current measurement / energy calculation is activated)*	
Current value / active power	never send
	send at intervals
	send on change
	send at intervals and on change
Minutes *(visible if sending at intervals)*	0 – 59 **[10]**
Seconds *(visible if sending at intervals)*	0 – 59 **[0]**
Send on change by *(visible if sending on change)*	1 mA / 0.04 W
	5 mA / 0.2 W
	25 mA / 1W
	50 mA / 2 W
	100 mA / 4 W
	200 mA / 8 W
	500 mA / 20 W
	1 A / 40 W
	2 A / 80 W
	3 A / 120 W
	4 A / 160 W
	5 A / 200 W
Energy value	**never send**
	send at intervals
	send on change
	send at intervals and on change
Hours *(visible if sending at intervals)*	0 – 24 **[24]**
Seconds *(visible if sending at intervals)*	0 – 59 **[0]**
Send on change by *(visible if sending on change)*	1 kWh
	5 kWh
	10 kWh
Reset current energy value by ETS download or object Reset *(visible if sending at intervals and/or on change)* *(Note: Reset function/object must be activated in the basic settings.)*	**deactivated**
	activated

No.	Name	Function	C	R	W	T	U
10	R1: Output (DPT 14.056)	Active power (in W)	C	–	–	T	–
11	R1: Output (DPT 9.021)	Current (in mA)	C	–	–	T	–
12	R1: Input (DPT 1.015)	Reset energy value	C	–	W	–	–
13	R1: Output (DPT 13.013)	Energy value (in kWh)	C	–	–	T	–
241	General: Input (DPT 1.015)	Parameter reset	C	–	W	–	–

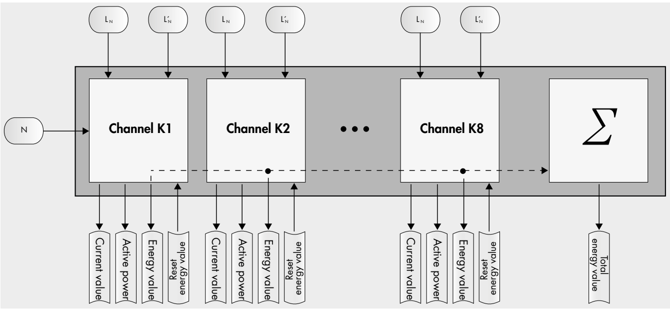
The actuator offers the option of calculating the total individual energy values of the channels. For this, energy calculation should be selected for the channels to be included in the calculation (parameter "Current measurement / Energy calculation" on the "General settings" sheet = "activated").

If a channel's energy value is reset, this is taken into account in the calculation of the total.

The total can be set up to be available as a status, i.e. only sent to the bus on request (e.g. visualisation). Alternatively, it can be sent periodically or when there is a change.

Base settings	
Delay in evaluation after relay closing in seconds	0 – 60 [10]
Total energy value	**Status**
	send at intervals
	send on change
	send at intervals and on change
Hours *(visible if sending at intervals)*	0 – 24 [10]
Minutes (*visible if sending at intervals)*	0 – 59 [0]
Send on change by *(visible if sending on change)*	1 kWh
	5 kWh
	10 kWh

No.	Name	Function	C	R	W	T	U
244	General: Output (DPT 13.013)	Reporting of total energy value (in kWh)	C	–	–	T	–



9.2 Current/Active power monitoring (adaptive/fixed)

By using current monitoring, the actuator monitors the current flowing when the relay is closed. By defining thresholds, it can be determined whether the current is too low or too high because of a faulty load. So it can be determined whether for example a certain number of lights in a lighting area have failed.

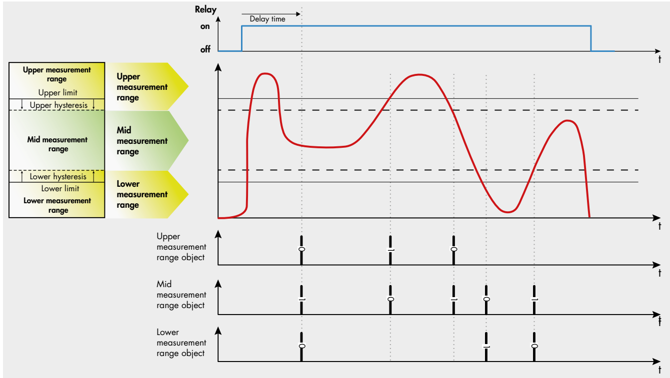
There is a choice of monitoring current value or active power.

The actuator offers several options for monitoring. With fixed monitoring, an upper limit, an upper hysteresis value, a lower limit and a lower hysteresis value are set.

The first step for adaptive monitoring is to set the adaptive behaviour in the parameters. Then for each of the upper limit, upper hysteresis value, lower hysteresis value and lower limit, the percentage deviation of these values from the learned value is given.

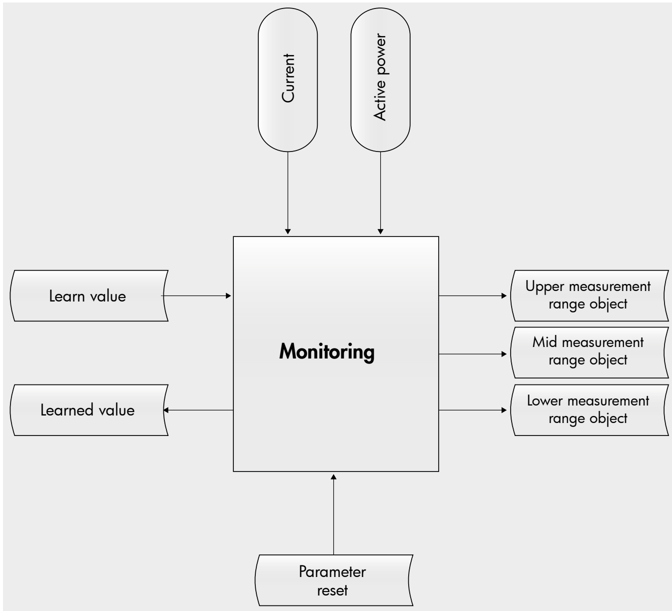
With both fixed and adaptive current monitoring, these values form the lower, mid and upper measurement ranges to be assigned to each object. You can now select which measurement range(s) are to be evaluated. Once evaluation of a measurement range has been activated, the sending behaviour of the object when entering the measurement range and when leaving it (taking hysteresis into account) can be set in the parameters (a 0 is sent, a 1 is sent, or there is no reaction).

With both adaptive and fixed current monitoring, a delay time after closing of the relay can be defined, in order to avoid evaluating a high inrush current for example, or fluctuations caused by relay bounce.



After the delay time finishes, all objects send their current status once, for which the parameter "When entering the ... measurement range send ... measurement range object" is evaluated.

Any instances of passing through the range when the relay is switched off are not evaluated.



9.2.1 Fixed monitoring

With fixed monitoring, an upper limit, an upper hysteresis value, a lower limit and a lower hysteresis value are set, from which ranges are formed.

R1: Fixed monitoring *(visible if Fixed monitoring is activated)*	
Monitoring	**of value of current**
	of active power
Delay in evaluation after relay closing in seconds	0 – 60 [10]
Evaluation of upper measurement range	**deactivated**
	activated
When entering upper measurement range sends upper measurement range object *(visible if upper measurement range is activated)*	"0"
	"1"
	no reaction

R1: Fixed monitoring *(visible if Fixed monitoring is activated)*	
When leaving upper measurement range sends upper measurement range object *(visible if upper measurement range is activated)*	**"0"**
	"1"
	no reaction
Upper current limit in milliamps *(visible with current monitoring)*	10 - 16000 **[12000]**
Hysteresis in milliamps *(visible with current monitoring)*	10 - 1000 **[500]**
Upper active power limit in Watts *(visible with monitoring of active power)*	10 - 16000 **[2400]**
Hysteresis in milliamps *(visible with monitoring of active power)*	10 - 1000 **[100]**
Evaluation of mid measurement range	**deactivated**
	activated
When entering mid measurement range sends mid measurement range object *(visible if mid measurement range is activated)*	"0"
	"1"
	no reaction
When leaving mid measurement range sends mid measurement range object *(visible if mid measurement range is activated)*	**"0"**
	"1"
	no reaction
Lower current limit in milliamps	0 - 16000 **[8000]**
Hysteresis in milliamps	10 - 1000 **[500]**
Lower active power limit in Watts *(visible with monitoring of active power)*	10 - 16000 **[1600]**
Hysteresis in milliamps *(visible with monitoring of active power)*	10 - 1000 **[100]**
Evaluation of lower measurement range	**deactivated**
	activated
When entering lower measurement range sends lower measurement range object *(visible if lower measurement range is activated)*	"0"
	"1"
	no reaction
When leaving lower measurement range sends lower measurement range object *(visible if lower measurement range is activated)*	**"0"**
	"1"
	no reaction

No.	Name	Function	C	R	W	T	U
16	R1: Output (DPT 1.002)	Upper measurement range object	C	–	–	T	–
17	R1: Output (DPT 1.002)	Mid measurement range object	C	–	–	T	–
18	R1: Output (DPT 1.002)	Lower measurement range object	C	–	–	T	–

9.2.2 Adaptive monitoring

The first step for adaptive monitoring is to set the adaptive behaviour in the parameters. Then for each of the upper limit, upper hysteresis value, lower hysteresis value and lower limit, the percentage deviation of these values from the learned value is given.

The learning process must take place when the relay is closed and during the learning event, the relay must remain closed. The relay state depends on the base functions, logic functions and the relay behaviour set in the parameters. To ensure that the relay is closed, the learning event can overwrite the priorities.

A learning event is started with a switch command. It can be selected whether this is an on or off command or either. The learning time can also be adjusted. The longer the measurement period, the more accurate the value. After the end of the learning event, the measured/learned value is sent.

A setting can be made for the learned current value to be overwritten by ETS download or object Reset. In this case, a value to be used as a starting value for the current will be given. Overwriting of the learned value can however be prevented by setting the parameter to "not overwriteable". The common "Parameter Reset" object (base settings), which sets the actuator back to its standard values, reverts to the defined value.

Note:

- After a download / After the first download, no value has been learned. Evaluation and telegrams are then invalid.

R1: Adaptive monitoring *(visible if adaptive monitoring is activated)*	
Monitoring	**of value of current**
	of active power
Delay in evaluation after relay closing in seconds	0 – 60 **[10]**
Object value at start of learning event	"0"
	"1"
	"0" / "1"
Learning time in seconds	0 -240 **[60]**
Learning event overrides priorities	deactivated
	activated
Learnt current value by ETS download or object Reset *(visible on "Monitoring of value of current")*	overwriteable
	not overwriteable
Value of current in milliamps *(visible if overwriteable)*	10 – 16000 **[10000]**
Learnt active power value by ETS download or object Reset *(visible on "Monitoring of active power")* *(Note: Reset function/object must be activated in the base settings.)*	overwriteable
	not overwriteable
Active power in Watts *(visible if overwriteable)*	1 – 16000 **[2000]**
Evaluation of upper measurement range	**deactivated**
	activated
When entering upper measurement range sends upper measurement range object *(visible if upper measurement range is activated)*	"0"
	"1"
	no reaction

R1: Adaptive monitoring *(visible if adaptive monitoring is activated)*	
When leaving upper measurement range sends upper measurement range object *(visible if upper measurement range is activated)*	**"0"**
	"1"
	no reaction
Upper current limit in % of learned value *(visible with "Monitoring of value of current")*	101 - 200 **[120]**
Hysteresis in % of upper current limit *(visible with "Monitoring of value of current")*	1 - 100 **[5]**
Upper active power limit in % of learned value *(visible with "Monitoring of active power")*	101 - 200 **[120]**
Hysteresis in % of upper active power limit *(visible with "Monitoring of active power")*	1 - 100 **[5]**
Evaluation of mid measurement range	**deactivated**
	activated
When entering mid measurement range sends mid measurement range object *(visible if mid measurement range is activated)*	"0"
	"1"
	no reaction
When leaving mid measurement range sends mid measurement range object *(visible if mid measurement range is activated)*	**"0"**
	"1"
	no reaction
Lower current limit in % of learned value *(visible with "Monitoring of value of current")*	0 - 99 **[80]**
Hysteresis in % of lower current limit *(visible with "Monitoring of value of current")*	1 - 100 **[5]**
Lower active power limit in % of learned value *(visible with "Monitoring of active power")*	10 - 99 **[80]**
Hysteresis in % of lower active power limit *(visible with "Monitoring of active power")*	1 - 100 **[5]**
Evaluation of lower measurement range	**deactivated**
	activated
When entering lower measurement range sends lower measurement range object *(visible if lower measurement range is activated)*	"0"
	"1"
	no reaction
When leaving lower measurement range sends lower measurement range object *(visible if lower measurement range is activated)*	**"0"**
	"1"
	no reaction

No.	Name	Function	C	R	W	T	U
15	R1: Input (DPT 1.017)	Learn current value	C	–	W	–	–
15	R1: Input (DPT 1.017)	Learn active power value	C	–	W	–	–
16	R1: Output (DPT 1.002)	Upper measurement range object	C	–	–	T	–
17	R1: Output (DPT 1.002)	Mid measurement range object	C	–	–	T	–
18	R1: Output (DPT 1.002)	Lower measurement range object	C	–	–	T	–
19	R1: Output (DPT 9.021)	Learned current value	C	–	–	T	–
19	R1: Output (DPT 14.056)	Learned active power	C	–	–	T	–
241	General: Input (DPT 1.015)	Parameter reset	C	–	W	–	–

9.3 Service hours counter

Using the service hours counter, a channel's service hours can be monitored. For this, the state to be monitored must first be defined. The most common use case is how long the connected load is switched on. For this, the duration that the relay is closed must be measured. Alternatively, a current or active power limit can be used, which must be exceeded. In this case, only the duration for which a load was actually connected is counted.

The duration for which the relay is open can also be measured, or the duration below a minimal current/active power limit.

In order to determine the service hours of the switch actuator itself, it can be set in the parameters that both relay states (closed and open) should be monitored. However, this setting must only be selected for one channel, and the other channels can if required monitor the relay states. In this way, using one channel to monitor the relay states "closed or open", the service hours count of the switch actuator can be determined, while for the remaining channels, the service hours count of the connected loads is determined.

The service hours counter internally counts up the hours, and this value is continuously compared with the service hours limit. The service hours limit can be set in the parameters from 0 to 100,000 hours. When in service, this value can also be changed by a 2-byte object via the bus. This resets the current service hours counter. If this limit is reached, the channel sends a message. This can be a "1" or a "0" telegram.

If the service hours counter has reached the service hours limit and has reported this to the bus, the service hours counter has to be cleared manually. Only then does the event start afresh. Service hours continue to be counted after the service hours limit has been reached.

The current service hours status can be sent when there is a change. The size of the change can be selected between 1 and 24 hours.

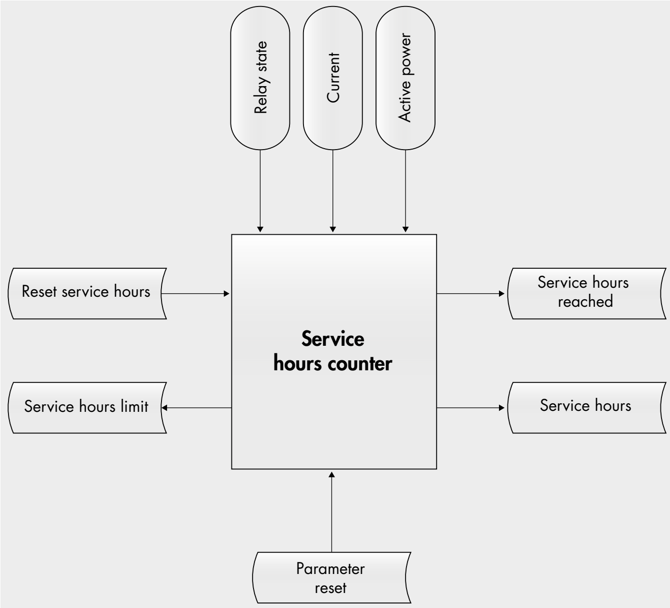
The reporting object (Service hours reached) can be sent periodically. But also, only a change can be sent (minimising bus load).

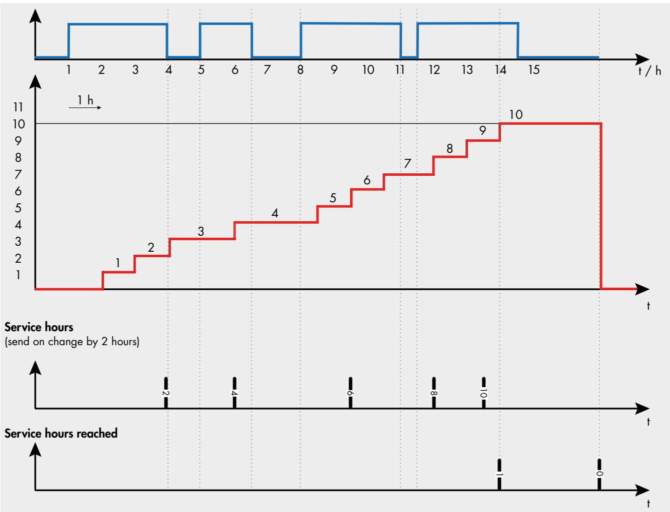
When the application is first loaded, the service hours limit stored in the parameters is stored in the actuator and the service hours counter is set to zero.

During operation, the service hours counter changes, and the limit can be changed by a 2-byte telegram. When the application is reloaded, you can decide whether the current values should be overwritten by the ETS or not.

The "Parameter Reset" object (base settings) resets the actuator to defined values. For the service hours counter, this is the stored service hours limit.

If bus voltage is lost, the current value of the service hours counter is saved. On bus voltage recovery, it is reinstated.





R1: Service hours counter *(visible if Service hours is activated)*	
Service hours counter	**Detection via relay state**
	Detection via value of current
	Detection via active power
Relay state in which counting should take place *(visible on "Detection via relay state")*	**closed**
	open
	closed or open
Counting should take place when *(visible on "Detection via value of current" or "Detection via active power)*	**limit exceeded**
	below limit
Current limit in milliamps *(visible on "detection via value of current")*	0 – 16000 **[10000]**
Active power limit in Watts *(visible on "Detection via active power")*	0 – 16000 **[2000]**
Service hours limit in hours	0 - 100000 **[8760]**
Service hours limit	**determined by parameters**
	overwriteable by object

R1: Service hours counter *(visible if Service hours is activated)*	
Limit changed by object by ETS download or object Reset *(visible if "overwriteable by object" is activated)* *(Note: Reset function/object must be activated in the basic settings.)*	**overwriteable**
	not overwriteable
Current service hours status resettable by ETS download or object Reset *(Note: Reset function/object must be activated in the base settings.)*	**deactivated**
	activated
Service hours	**never send**
	send on change by
Hours	0 – 24 **[1]**
Reporting object, service hours limit reached	never send
	send on change
	send at intervals and on change
Hours *(visible if at intervals)*	0 – 24 **[24]**
Minutes *(visible if at intervals)*	0 – 59 **[0]**
Value of reporting object *(visible if Reporting object is sent)*	**"0" = not reached / "1" = reached**
	"1" = not reached / "0" = reached

No.	Name	Function	C	R	W	T	U
21	R1: Input (DPT 7.007)	Service hours limit	C	–	W	–	–
22	R1: Input (DPT 1.015)	Reset service hours	C	–	W	–	–
23	R1: Output (DPT 1.002)	Service hours reached	C	–	–	T	–
24	R1: Input (7.007)	Service hours	C	–	–	T	–
241	General: Input (DPT 1.015)	Parameter reset	C	–	W	–	–

9.4 Operating cycle counter

The functionality of the operating cycle counter is similar to that of the service hours counter. Instead of service hours, the number of operating cycles of the relay is counted. You can select whether only switch-on events, only switch-off events or both should be counted.

The operating cycles are counted and compared with an operating cycle limit value. If this limit is reached, a report (operating cycles reached) is created. This can be a "1" or a "0" telegram.

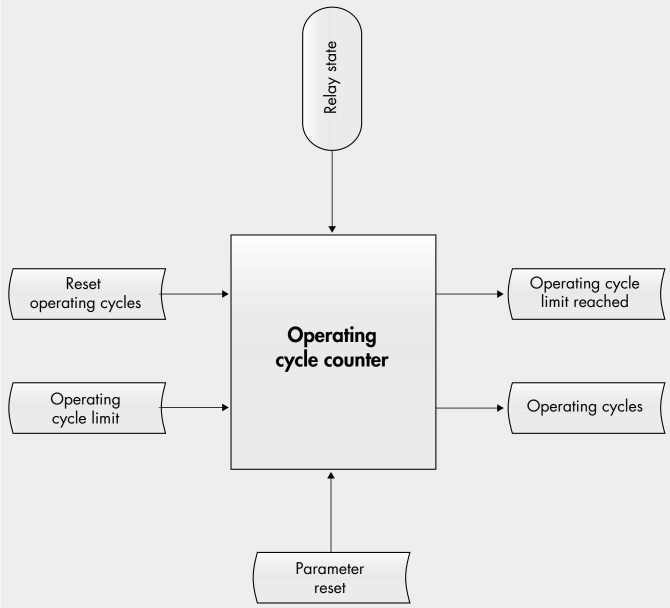
The limit can be defined by a parameter or can be changed in service by a 4-byte object (operating cycle limit). If a new value is sent via the object, the current counter status is reset.

The current operating cycle counter status (Operating cycle object) can be sent periodically. The reporting object (operating cycles reached) can also be sent periodically. But also, only a change can be sent (minimising bus load).

When the application is first loaded, the operating cycle limit stored in the parameters is stored in the actuator and the operating cycle counter is set to zero.

During operation, the operating cycle counter changes, and the limit can be changed by a 4-byte telegram. When the application is reloaded, you can decide whether the current values should be overwritten by the ETS or not.

If bus voltage is lost, the current value of the operating cycle counter is saved. On bus voltage recovery, it is reinstated.



R1: Operating cycle counter *(visible if Operating cycle counter is activated)*	
To be counted	switch-off events
	switch-on events
	switch-on and switch-off events
Operating cycle counter limit	0 - 100000 **[10000]**
Operating cycle counter limit	**determined by parameters**
	overwriteable by object
Limit changed by object by ETS download or object Reset *(visible if overwriteable by object)* *(Note: Reset function/object must be activated in the base settings.)*	**overwriteable**
	not overwriteable
Current operating cycle counter limit resettable by ETS download or object *(Note: Reset function/object must be activated in the base settings.)*	**deactivated**
	activated
Current counter status	never send
	send at intervals
	send on change
	send at intervals and on change
Hours *(visible if at intervals)*	0 – 24 **[24]**
Change up to sending *(visible if sending on change)*	10 - 10000 **[1000]**
Reporting object, operating cycle counter limit reached	never send
	send on change
	send at intervals and on change
Hours *(visible if at intervals)*	0 – 24 **[24]**
Minutes *(visible if at intervals)*	0 – 59 **[0]**
Value of reporting object *(visible if Reporting object is sent)*	**“0” = not reached / “1” = reached**
	“1” = not reached / “0” = reached

No.	Name	Function	C	R	W	T	U
26	R1: Input (DPT12.001)	Operating cycle limit	C	–	W	–	–

No.	Name	Function	C	R	W	T	U
27	R1: Input (DPT 1.015)	Reset operating cycles	C	–	W	–	–
28	R1: Output (DPT 1.002)	Operating cycles reached	C	–	–	T	–
29	R1: Output (DPT 12.001)	Operating cycles	C	–	–	T	–
241	General: Input (DPT 1.015)	Parameter reset	C	–	W	–	–