Application program description

µBrick Actuator Series

CONTENT

µBrick Actuator Series

1. Introduction

1.2 Using the application program

Product family: Actuators
Product type: Actuators
Manufacturer: IPAS GmbH

Name: µBrick actuator range
Order number:

Reference	Description	Order number
io66	6 inputs (bin & analog) / 6 resistive outputs	72130-180-01
o8	8 resistive outputs	72130-180-02
io66X	6 inputs (bin & analog) / 4 Resistive & 2 Capacitive outputs	72130-180-03
o12X	8 Resistive & 4 Capacitive outputs	72130-180-04
o18	8 Resistive outputs	72130-180-05

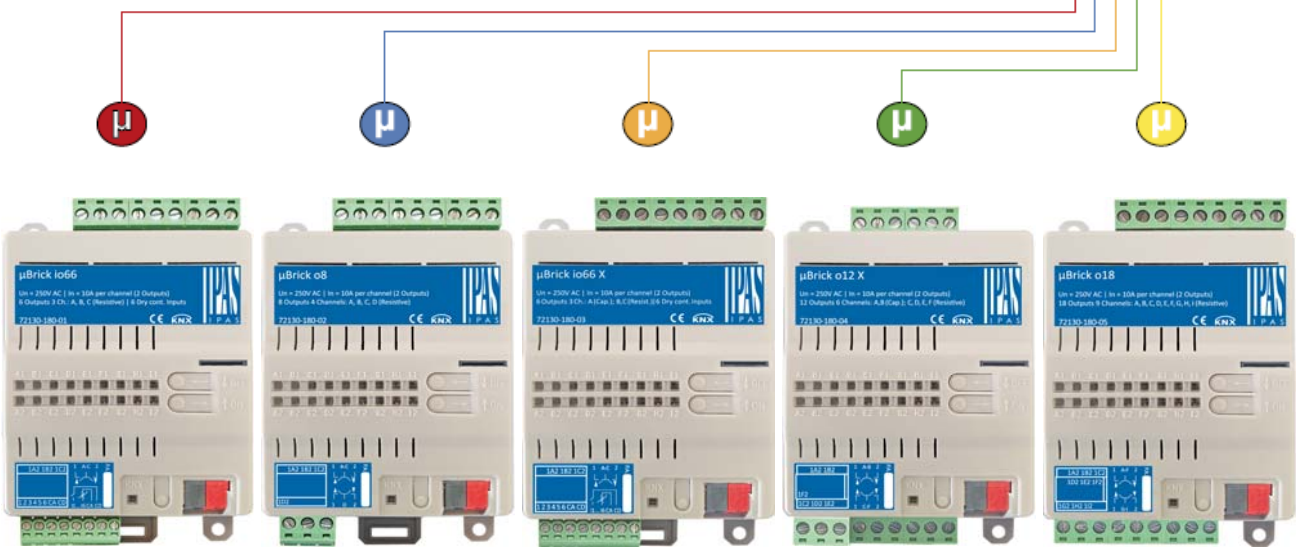
µBrick Actuator Series

1.3 General product information

The µBrick range consists of five different actuator types and distinguishes itself through its extremely small and compact construction. The application programs are built in such a way that basic functions can be projected intuitively. This basic functionality, however, can be greatly expanded by means of structured parameter menus in the ETS.

1.3.1 µBrick range

Mounting type		Name	Output Type	DIN MOD	Inputs	Outputs
DIN and Flush Mount	Very Flat	µBrick io66	6R 10A	4	6	6
		µBrick o8	8R 10A	4	0	8
	Flat	µBrick io66X	2C+4R 10A	4	6	6
		µBrick o12X	4C+8R 10A	4	0	12
		µBrick o18	18R 10A	4	0	18



µBrick Actuator Series

1.3.2 Decentralized installation

µBrick actuators are installed on either a standard DIN rail or directly on the installation base. These flexible mounting options in connection with the extremely small size offer a wide range of installation options. With µBrick actuators, the KNX installation is decentralized and no longer requires a distribution box.

Decentralized installations reduce the thermal load because of reduced cable length. A lower thermal load makes buildings safer. Each decentralized device reduces the space required in the installation. Small distributions are cheaper and take up less room. These are significant advantages both for commercial and domestic buildings.
µBrick actuators can be installed in flush-mounted cabinets with a minimum size of 150mm x 80mm x40mm (LXBXT). The mounting depth of just 40mm is significantly less than the mounting depth of a common switch box, which is 55mm.
Such a low depth makes flush mounting very easy. Individual switch functions can be performed directly in the room where they are required. Of course, µBrick actuators can also be wall mounted. In commercial buildings they are therefore perfect for mounting in cavity walls and floors - again with a minimum amount of effort.

1.3.3 General properties of the ETS application program

1.3.3.1 Installing the application program

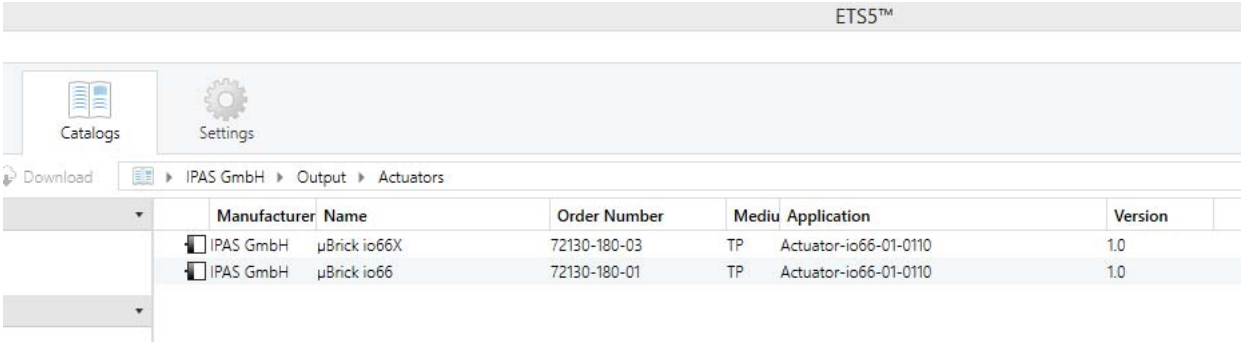
The application for the µBrick ACTUATOR RANGE is based on a powerful KNX communications stack of the System-B type, with up to 1000 KNX objects. It is designed as a standard ETS application program and no plug-in for ETS-3 and ETS-4 is needed.
After the import the product can be integrated as usual into the ETS.

ETS application names:

Actuator-io66-01-0110
Actuator-o8-01-0110
Actuator-o12-01-0110
Actuator-o18-01-0110

µBrick Actuator Series

It can be found under product family “Output” and product type “Actuators”.



1.3.4 Preliminary basic concepts

Output: channel type selection
In the µBrick actuator range, each channel is composed of two mechanical outputs (relays):

- If the channel type is selected to be a “Binary” output, then you will have two totally independent outputs in the Application program.
- On the contrary, if you select the channel type to be “Shutter/Blind”, then these two outputs work as one shutter/blind channel. The first relay will be for movement UP and second one for movement DOWN.

Type of contact
It is possible to select the type of contact to be normally open or normally closed, which is a common feature of modern actuators. It is very important though to keep in mind that these terms only refer to the mechanical contact.

On the other hand, in this application program the terms ON and OFF will be frequently used, whereas ON is always = “1” and OFF is always = “0”. Independent from the type of contact (NO/NC), if you send an ON (“1”) to the switching object, the status object will always send an ON (“1”); and vice versa.

- NO-Normally open (ON=close, OFF=open): the output relay closes with ON (“1”) and opens with OFF (“0”).
- NC-Normally close (ON=open, OFF=close): the output relay closes with OFF (“0”) and opens with ON (“1”).

Maximum sending speed
Should an output object be changed faster than the maximum sending speed of the KNX stack, these changes will be ignored and only the last change will be sent to the bus.

Cyclical sending
The application program contains multiple occasions where cyclic sending for different functions can be used. When this function is activated, the corresponding object will not send the telegram once, but repeat it infinitely.

Frequency and time calculation
The calculation of the preferred time (cyclical sending, delays, staircase, etc.) is done by multiplying the “time Base” by the “time Factor”.

µBrick Actuator Series

Selection of data point type
During the configuration of the actuator, you will be asked to choose the data point type. It is very important to correctly define the DPT because this will change the size and type of the object; also, the data will be differently interpreted. E.g.: 1 Byte counter value = 0 to 255, whereas 1 Byte scaling value = 0 to 100%.

Additional/advanced functions (channel related)
In order to keep the application program as easy as possible, only the main and most important functions are displayed at first sight. You will often find the possibility to activate the Additional or Advanced Functions, which disclose new functions that are not essential, but can be very useful.
Also, see General_Settings_Advanced_Functions.

Scenes
In this actuator range we can find two types of Scenes:
- KNX Scenes: fully KNX standard 1 byte scenes.
- Advanced Scenes controller (not available in Outputs): free configurable trigger conditions (start, save, stop and restore) and scene actions with time delays.

Enable/disable object
Most of the actuator's modules can be deactivated with a "… disable" object. The value (1 or 0) used to disable can also be configured.
This option can be very useful for many reasons, including simplifying the configuration: for instance, the logic functions might be a complex task that can take a while to finish; in the meantime, you don't want these modules to be active and cause unwanted actions. Therefore, you can disable them until you finish programming. Another example: you can simply activate/deactivate the timers for the irrigation system when not needed.

End-user parameters
It is very important for the end user to be able to change (via dedicated objects linked, for instance, to a visualization) certain settings of his/her KNX installation. This actuator allows for these changes to be maintained even when downloading the application program again. In "overwrite end-user parameter values at download" you will find an in-depth explanation on when and how to overwrite/maintain the changes made by the end-user.

µBrick Actuator Series

2. ETS communication objects overview

The µBrick actuators communicates via the KNX bus based on powerful communication stacks. Altogether 998 communication objects for the o18 (depending of the device model) are available for the communication.

GENERAL OBJECTS & ADVANCED FUNCTIONS

µBrick Actuator Series

	N..	Name	Object Function	Le...	Data Type	...	R	W	T	U	Priority
	1	Central switching	< On / Off	1 bit	1-bit	C	-	W	-	-	Low
	2	Central move	< Up/Down/Position	1 bit	1-bit	C	-	W	-	-	Low
	3	Central cyclic telegram for monitoring	> Cyclic ON telegrams	1 bit	1-bit	C	R	-	T	-	Low
	4	Telegram at bus recovery	> Sends parameterized value	1 bit	1-bit	C	-	-	T	-	Low
	5	Manual control disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	-	-	Low
	6	Alarm 1	< 2 bytes float	2 Byte	2-byte float value	C	R	W	-	-	Low
	14	Alarm ACK	< Ack. with 1	1 bit	1-bit	C	-	W	-	-	Low
	15	Alarm 1 setpoint	< 2 bytes float	2 Byte	2-byte float value	C	R	W	-	-	Low
	23	Alarm 1 hysteresis	< 2 bytes float	2 Byte	2-byte float value	C	R	W	-	-	Low
	31	Alarm 1 disable	< Disable = 1 / Enable = 0	1 bit	1-bit	C	R	W	-	-	Low
	39	Logic 1 disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	-	-	Low
	40	Logic 1 input 1	< On / Off	1 bit	1-bit	C	R	W	T	U	Low
	41	Logic 1 input 2	< On / Off	1 bit	1-bit	C	R	W	T	U	Low
	42	Logic 1 input 3	< On / Off	1 bit	1-bit	C	R	W	T	U	Low
	43	Logic 1 input 4	< On / Off	1 bit	1-bit	C	R	W	T	U	Low
	44	Logic 1 output	> 1 byte unsigned	1 Byte	counter pulses (0..255)	C	R	-	T	-	Low
	159	Scene 1 input	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	counter pulses (0..255)	C	-	W	-	-	Low
	160	Scene 1 disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	-	-	Low
	161	Scene 1 event 1	> On / Off	1 bit	1-bit	C	-	W	T	U	Low
	162	Scene 1 event 2	> 0..100%	1 Byte	percentage (0..100%)	C	-	W	T	U	Low
	163	Scene 1 event 3	> 1byte unsigned	1 Byte	counter pulses (0..255)	C	-	W	T	U	Low
	164	Scene 1 event 4	> 2 bytes unsigned	2 Byte	pulses	C	-	W	T	U	Low
	165	Scene 1 event 5	> 2 bytes float	2 Byte	2-byte float value	C	-	W	T	U	Low
	166	Scene 1 event 6	> 4 bytes unsigned	4 Byte	counter pulses (unsigned)	C	-	W	T	U	Low
	167	Scene 1 event 7	> 4 bytes float	4 Byte	4-byte float value	C	-	W	T	U	Low
	168	Scene 1 event 8	> 4 bytes signed	4 Byte	counter pulses (signed)	C	-	W	T	U	Low
	259	Advanced Scene 1 input	< 2 bytes float	2 Byte	2-byte float value	C	-	W	-	-	Low
	260	Advanced Scene 1 disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	-	-	Low
	261	Advanced Scene 1 event 1	<> On / Off	1 bit	1-bit	C	-	W	T	U	Low
	262	Advanced Scene 1 event 2	<> 0..100%	1 Byte	percentage (0..100%)	C	-	W	T	U	Low
	263	Advanced Scene 1 event 3	<> 1byte unsigned	1 Byte	counter pulses (0..255)	C	-	W	T	U	Low
	264	Advanced Scene 1 event 4	<> 2 bytes unsigned	2 Byte	pulses	C	-	W	T	U	Low
	265	Advanced Scene 1 event 5	<> 2 bytes float	2 Byte	2-byte float value	C	-	W	T	U	Low
	266	Advanced Scene 1 event 6	<> 4 bytes unsigned	4 Byte	counter pulses (unsigned)	C	-	W	T	U	Low
	267	Advanced Scene 1 event 7	<> 4 bytes float	4 Byte	4-byte float value	C	-	W	T	U	Low
	268	Advanced Scene 1 event 8	<> 2 bytes signed	2 Byte	pulses difference	C	-	W	T	U	Low
	359	Timer 1 trigger	< 2 bytes float	2 Byte	2-byte float value	C	-	W	-	-	Low
	362	Timer 1 disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	-	-	Low
	363	Timer 1 output	> 2 bytes float	2 Byte	2-byte float value	C	-	-	T	-	Low
	409	Setpoint 1 output regulator	> On / Off	1 bit	1-bit	C	R	-	T	-	Low
	410	Setpoint 1 setpoint value/status	<> 2 bytes float	2 Byte	2-byte float value	C	R	W	T	-	Low
	411	Setpoint 1 Heat / Cool	< Heat = 1 / Cool = 0	1 bit	1-bit	C	R	W	-	-	Low
	412	Setpoint 1 input ext. sensor value	< 2 bytes float	2 Byte	2-byte float value	C	R	W	-	-	Low
	413	Setpoint 1 disable	< On / Off	1 bit	1-bit	C	R	W	-	-	Low

µBrick Actuator Series

BINARY OUTPUT CHANNEL & INPUT

559	[A1] Switching On / Off	< On / Off	1 bit	switch	C	-	W	-	-	Low
560	[A1] Switching toggle/inverted	< Toggle with 0 and 1	1 bit	switch	C	-	W	-	-	Low
561	[A1] Switching status	> On / Off	1 bit	switch	C	R	-	T	-	Low
562	[A1] RunHour counter value	> 4 bytes unsigned	4 Byte	counter pulses (unsigned)	C	R	-	T	-	Low
563	[A1] RunHour counter threshold	< Reading/writing threshold	4 Byte	counter pulses (unsigned)	C	R	W	T	-	Low
564	[A1] RunHour counter alarm	> 1 = Alarm, 0 = No alarm	1 bit	1-bit	C	R	-	T	-	Low
565	[A1] RunHour counter reset	< 1 = Reset, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
566	[A1] RunHour counter value at reset	> 4 bytes unsigned	4 Byte	counter pulses (unsigned)	C	R	-	T	-	Low
567	[A1] Switching counter value	> 4 bytes unsigned	4 Byte	counter pulses (unsigned)	C	R	-	T	-	Low
568	[A1] Switching counter threshold	< Reading/writing threshold	4 Byte	counter pulses (unsigned)	C	R	W	T	-	Low
569	[A1] Switching counter alarm	> 1 = Alarm, 0 = No alarm	1 bit	1-bit	C	R	-	T	-	Low
570	[A1] Switching counter reset	< 1 = Reset, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
571	[A1] Switching counter value at reset	> 4 bytes unsigned	4 Byte	counter pulses (unsigned)	C	R	-	T	-	Low
572	[A1] Scene number	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	counter pulses (0..255)	C	-	W	-	-	Low
573	[A1] Scene disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	-	-	Low
574	[A1] Timer 1 trigger	< On / Off	1 bit	1-bit	C	-	W	-	-	Low
575	[A1] Timer 1 change staircase factor	< 1 byte unsigned	1 Byte	counter pulses (0..255)	C	R	W	-	-	Low
576	[A1] Timer 1 warning pulse	> On / Off	1 bit	switch	C	R	-	T	-	Low
577	[A1] Timer 1 disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	T	-	Low
578	[A1] Timer 2 trigger	< On / Off	1 bit	1-bit	C	-	W	-	-	Low
579	[A1] Timer 2 change staircase factor	< 1 byte unsigned	1 Byte	counter pulses (0..255)	C	R	W	-	-	Low
580	[A1] Timer 2 warning pulse	> On / Off	1 bit	switch	C	R	-	T	-	Low
581	[A1] Timer 2 disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	T	-	Low
582	[A1] Disable channel	< On / Off	1 bit	1-bit	C	R	W	T	-	Low
583	[A2] Switching On / Off	< On / Off	1 bit	switch	C	-	W	-	-	Low
585	[A2] Switching status	> On / Off	1 bit	switch	C	R	-	T	-	Low
704	[In1] Switching short	> On / Off	1 bit	switch	C	R	W	T	-	Low
705	[In1] Switching long	> On / Off	1 bit	switch	C	R	W	T	-	Low
744	[In1] Alarm short circuit	> Alarm = 1, No alarm = 0	1 bit	1-bit	C	R	-	T	-	Low
745	[In1] Alarm open circuit	> Alarm = 1, No alarm = 0	1 bit	1-bit	C	R	-	T	-	Low
746	[In1] Alarm open / short circuit	> Alarm = 1, No alarm = 0	1 bit	1-bit	C	R	-	T	-	Low
747	[In1] Monitor input ACK	< Ack. with 1	1 bit	1-bit	C	R	W	-	-	Low
961	Alarm 1 status	> ON = Alarm, OFF = No alarm	1 bit	1-bit	C	R	-	T	-	Low

µBrick Actuator Series

SHUTTER OUTPUT CHANNEL

	N..	Name	Object Function	Le...	Data Type	...	R	W	T	U	Priority
	559	[A] Move	< 0=up/1=down	1 bit	up/down	C	-	W	-	-	Low
	560	[A] Stop (Blind=Stop/step)	< 0=stop/step, 1=stop/step	1 bit	1-bit	C	-	W	-	-	Low
	561	[A] Move to position	< 0..100%	1 Byte	percentage (0..100%)	C	-	W	-	-	Low
	562	[A] Move slat	< 0..100%	1 Byte	percentage (0..100%)	C	-	W	-	-	Low
	563	[A] Change upper limit	<> 0..100%	1 Byte	percentage (0..100%)	C	R	W	T	-	Low
	564	[A] Change lower limit	<> 0..100%	1 Byte	percentage (0..100%)	C	R	W	T	-	Low
	565	[A] Status blind position	> 0..100%	1 Byte	percentage (0..100%)	C	R	-	T	-	Low
	566	[A] Status blind lower end position	> 1 = Totally down / 0 = not	1 bit	1-bit	C	R	-	T	-	Low
	567	[A] Status blind upper end position	> 1 = Totally up / 0 = not	1 bit	1-bit	C	R	-	T	-	Low
	568	[A] Status slat position	> 0..100%	1 Byte	percentage (0..100%)	C	R	-	T	-	Low
	569	[A] Preset 1 execute	< 1 = Execute, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	570	[A] Preset 2 execute	< 1 = Execute, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	571	[A] Preset 3 execute	< 1 = Execute, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	572	[A] Preset 4 execute	< 1 = Execute, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	573	[A] Preset 1 change move position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	574	[A] Preset 2 change move position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	575	[A] Preset 3 change move position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	576	[A] Preset 4 change move position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	577	[A] Preset 1 change slat position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	578	[A] Preset 2 change slat position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	579	[A] Preset 3 change slat position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	580	[A] Preset 4 change slat position	< 0..100%	1 Byte	percentage (0..100%)	C	R	W	-	-	Low
	581	[A] Preset 1 save	< 1 = Save, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	582	[A] Preset 2 save	< 1 = Save, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	583	[A] Preset 3 save	< 1 = Save, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	584	[A] Preset 4 save	< 1 = Save, 0 = Nothing	1 bit	1-bit	C	-	W	-	-	Low
	585	[A] Scene number	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	counter pulses (0..255)	C	-	W	-	-	Low
	586	[A] Scene disable	< Disable = 0 / Enable = 1	1 bit	1-bit	C	R	W	-	-	Low
	587	[A] Disable channel	< On / Off	1 bit	1-bit	C	R	W	T	-	Low
	588	[A] Move inverted	< 1=up/0=down	1 bit	1-bit	C	-	W	-	-	Low
	589	[A] Disable limits / calibrate	< Disable =0 / En&calibrate =1	1 bit	1-bit	C	R	W	-	-	Low

µBrick Actuator Series

	Text	Function text	Object Size	Flags	Datapoint type
1	Central switching	< On / Off	1 Bit	-WC---	[1] 1.xxx
Each and every channel can individually be configured to have no reaction, switch ON / OFF or start the timer 1 reaction at on when this object receives a parametrized value. See parameter description to see all possibilities.					
1	Central switching/move blind	< On / Off, Up/Down/Position	1 Bit	-WC---	[1] 1.xxx
Each and every channel can individually be configured to have no reaction, switch ON / OFF or start the timer 1 reaction at on, move UP/DOWN or move to a specific position when this object receives a parametrized value. See parameter description to see all possibilities.					
2	Central move	< Up/Down/Position	1 Bit	-WC---	[1] 1.xxx
Each and every channel can individually be configured to have no reaction, move UP/DOWN or move to a specific position when this object receives a parametrized value. See parameter description to see all possibilities.					
3	Central cyclic telegram for monitoring	> Cyclic ON telegrams	1 Bit	R-CT--	[1] 1.xxx
This object sends an ON telegram cyclic with bus voltage. This can be used to supervise a bus line. A channel in the mainline with a staircase timer can be triggered with a higher frequency than the staircase time by this object. Should the line fail the staircase will expire and therefore the "Line status light" will switch OFF.					
4	Telegram at bus recovery	> Sends parameterized value	1 Bit	--CT--	[1] 1.xxx
This object will send a parametrized value to the bus after bus voltage return. This can be used to trigger an event, like a scene to set up the whole installation at bus return.					
4	Telegram at bus recovery	> Sends parameterized value	1 Byte	--CT--	[5.10] DPT_Value_1_Ucount
This object will send a parametrized value to the bus after bus voltage return. This can be used to trigger an event, like a scene to set up the whole installation at bus return.					
4	Telegram at bus recovery	> Sends parameterized value	1 Byte	--CT--	[5.1] DPT_Scaling
This object will send a parametrized value to the bus after bus voltage return. This can be used to trigger an event, like a scene to set up the whole installation at bus return.					
4	Telegram at bus recovery	> Sends parameterized value	2 Bytes	--CT--	[9] 9.xxx
This object will send a parametrized value to the bus after bus voltage return. This can be used to trigger an event, like a scene to set up the whole installation at bus return.					
5	Manual control disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The manual buttons on the device can be deactivated by this object like this: Disable = 1 / Enable = 0					
5	Manual control disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The manual buttons on the device can be deactivated by this object like this: Disable = 0 / Enable = 1					
6	Alarm 1	< On / Off	1 Bit	RWC--I	[1] 1.xxx

µBrick Actuator Series

This object is the alarm 1 trigger object. In the parameters one can define with which value it should be in the alarm state.					
6	Alarm 1	< 0..100%	1 Byte	RWC--I	[5.1] DPT_Scaling
This object is the alarm 1 trigger object. In the parameters one can define with which value it should be in the alarm state.					
6	Alarm 1	< 1 byte unsigned	1 Byte	RWC--I	[5.10] DPT_Value_1_Ucount
This object is the alarm 1 trigger object. In the parameters one can define with which value it should be in the alarm state.					
6	Alarm 1	< 2 bytes float	2 Bytes	RWC--I	[9] 9.xxx
This object is the alarm 1 trigger object. In the parameters one can define with which value it should be in the alarm state.					
6	Alarm 1	< 4 bytes unsigned	4 Bytes	RWC--I	[12.1] DPT_Value_4_Ucount
This object is the alarm 1 trigger object. In the parameters one can define with which value it should be in the alarm state.					
6	Alarm 1	< 4 bytes float	4 Bytes	RWC--I	[14] 14.xxx
This object is the alarm 1 trigger object. In the parameters one can define with which value it should be in the alarm state.					
14	Alarm ACK	< Ack. with 0	1 Bit	-WC---	[1] 1.xxx
When activating the acknowledge function this object appears. This is to acknowledge the alarm by sending a 0 to this object. Alarms can only be acknowledged if the alarm has disappeared					
14	Alarm ACK	< Ack. with 1	1 Bit	-WC---	[1] 1.xxx
When activating the acknowledge function this object appears. This is to acknowledge the alarm by sending a 1 to this object. Alarms can only be acknowledged if the alarm has disappeared					
15	Alarm 1 setpoint	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
If the alarm is configured to be an analog alarm then the threshold of this alarm can be set by this object					
15	Alarm 1 setpoint	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
If the alarm is configured to be an analog alarm then the threshold of this alarm can be set by this object					
15	Alarm 1 setpoint	< 2 bytes float	2 Bytes	RWC---	[9] 9.xxx
If the alarm is configured to be an analog alarm then the threshold of this alarm can be set by this object					
15	Alarm 1 setpoint	< 4 bytes unsigned	4 Bytes	RWC---	[12.1] DPT_Value_4_Ucount

µBrick Actuator Series

If the alarm is configured to be an analog alarm then the threshold of this alarm can be set by this object					
15	Alarm 1 setpoint	< 4 bytes float	4 Bytes	RWC---	[14] 14.xxx
If the alarm is configured to be an analog alarm then the threshold of this alarm can be set by this object					
23	Alarm 1 hysteresis	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
If the alarm is configured to be an analog alarm then the hysteresis of this alarm setpoint can be changed by this object					
23	Alarm 1 hysteresis	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
If the alarm is configured to be an analog alarm then the hysteresis of this alarm setpoint can be changed by this object					
23	Alarm 1 hysteresis	< 2 bytes float	2 Bytes	RWC---	[9] 9.xxx
If the alarm is configured to be an analog alarm then the hysteresis of this alarm setpoint can be changed by this object					
23	Alarm 1 hysteresis	< 4 bytes float	4 Bytes	RWC---	[14] 14.xxx
If the alarm is configured to be an analog alarm then the hysteresis of this alarm setpoint can be changed by this object					
23	Alarm 1 hysteresis	< 4 bytes unsigned	4 Bytes	RWC---	[12.1] DPT_Value_4_Ucount
If the alarm is configured to be an analog alarm then the hysteresis of this alarm setpoint can be changed by this object					
31	Alarm 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The alarm can be disabled by sending a 1 to this object.					
39	Logic 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The logic function can be disabled by sending a 0					
39	Logic 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The logic function can be disabled by sending a 1					
40	Logic 1 input 1	< On / Off	1 Bit	RWCTU-	[1] 1.xxx
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 0..100%	1 Byte	RWCTU-	[5.1] DPT_Scaling

µBrick Actuator Series

This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 1 byte signed	1 Byte	RWCTU-	[6.10] DPT_Value_1_Count
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 1 byte unsigned	1 Byte	RWCTU-	[5.10] DPT_Value_1_Ucount
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 2 bytes unsigned	2 Bytes	RWCTU-	[7.1] DPT_Value_2_Ucount
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 2 bytes float	2 Bytes	RWCTU-	[9] 9.xxx
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 2 bytes signed	2 Bytes	RWCTU-	[8.1] DPT_Value_2_Count
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 4 bytes unsigned	4 Bytes	RWCTU-	[12.1] DPT_Value_4_Ucount
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 4 bytes float	4 Bytes	RWCTU-	[14] 14.xxx
This is the first of 4 logic inputs of this logic block					
40	Logic 1 input 1	< 4 bytes signed	4 Bytes	RWCTU-	[13.1] DPT_Value_4_Count
This is the first of 4 logic inputs of this logic block					
41	Logic 1 input 2	< On / Off	1 Bit	RWCTU-	[1] 1.xxx
This is the second of 4 logic inputs of this logic block					
41	Logic 1 Enable / Disable Gate	< Disable = 1 / Enable = 0	1 Bit	RWCT--	[1] 1.xxx
If the logic function is configured to be a Gate function then this input is used to enable or disable the gate. When the gate is disabled the input will not be sent to the output. This object can also be used to trigger the input to the output with different conditions (please see the parameter description to see al possibilities)					

µBrick Actuator Series

41	Logic 1 Enable / Disable Gate	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1] 1.xxx
If the logic function is configured to be a Gate function then this input is used to enable or disable the gate. When the gate is disabled the input will not be sent to the output. This object can also be used to trigger the input to the output with different conditions (please see the parameter description to see al possibilities)					
41	Logic 1 input 2	< 1 byte signed	1 Byte	RWCTU-	[6.10] DPT_Value_1_Count
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 0..100%	1 Byte	RWCTU-	[5.1] DPT_Scaling
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 1 byte unsigned	1 Byte	RWCTU-	[5.10] DPT_Value_1_Ucount
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 2 bytes signed	2 Bytes	RWCTU-	[8.1] DPT_Value_2_Count
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 2 bytes unsigned	2 Bytes	RWCTU-	[7.1] DPT_Value_2_Ucount
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 2 bytes float	2 Bytes	RWCTU-	[9] 9.xxx
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 4 bytes unsigned	4 Bytes	RWCTU-	[12.1] DPT_Value_4_Ucount
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 4 bytes float	4 Bytes	RWCTU-	[14] 14.xxx
This is the second of 4 logic inputs of this logic block					
41	Logic 1 input 2	< 4 bytes signed	4 Bytes	RWCTU-	[13.1] DPT_Value_4_Count
This is the second of 4 logic inputs of this logic block					
42	Logic 1 input 3	< On / Off	1 Bit	RWCTU-	[1] 1.xxx

µBrick Actuator Series

This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 0..100%	1 Byte	RWCTU-	[5.1] DPT_Scaling
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 1 byte unsigned	1 Byte	RWCTU-	[5.10] DPT_Value_1_Ucount
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 1 byte signed	1 Byte	RWCTU-	[6.10] DPT_Value_1_Count
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 2 bytes unsigned	2 Bytes	RWCTU-	[7.1] DPT_Value_2_Ucount
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 2 bytes signed	2 Bytes	RWCTU-	[8.1] DPT_Value_2_Count
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 2 bytes float	2 Bytes	RWCTU-	[9] 9.xxx
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 4 bytes unsigned	4 Bytes	RWCTU-	[12.1] DPT_Value_4_Ucount
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 4 bytes signed	4 Bytes	RWCTU-	[13.1] DPT_Value_4_Count
This is the third of 4 logic inputs of this logic block					
42	Logic 1 input 3	< 4 bytes float	4 Bytes	RWCTU-	[14] 14.xxx
This is the third of 4 logic inputs of this logic block					
43	Logic 1 input 4	< On / Off	1 Bit	RWCTU-	[1] 1.xxx
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 0..100%	1 Byte	RWCTU-	[5.1] DPT_Scaling

µBrick Actuator Series

This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 1 byte unsigned	1 Byte	RWCTU-	[5.10] DPT_Value_1_Ucount
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 1 byte signed	1 Byte	RWCTU-	[6.10] DPT_Value_1_Count
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 2 bytes unsigned	2 Bytes	RWCTU-	[7.1] DPT_Value_2_Ucount
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 2 bytes signed	2 Bytes	RWCTU-	[8.1] DPT_Value_2_Count
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 2 bytes float	2 Bytes	RWCTU-	[9] 9.xxx
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 4 bytes signed	4 Bytes	RWCTU-	[13.1] DPT_Value_4_Count
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 4 bytes float	4 Bytes	RWCTU-	[14] 14.xxx
This is the fourth of 4 logic inputs of this logic block					
43	Logic 1 input 4	< 4 bytes unsigned	4 Bytes	RWCTU-	[12.1] DPT_Value_4_Ucount
This is the fourth of 4 logic inputs of this logic block					
44	Logic 1 output	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 1 byte signed	1 Byte	R-CT--	[6.10] DPT_Value_1_Count
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 2 bytes signed	2 Bytes	R-CT--	[8.1] DPT_Value_2_Count
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 4 bytes signed	4 Bytes	R-CT--	[13.1] DPT_Value_4_Count
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
44	Logic 1 output	> 4 bytes float	4 Bytes	R-CT--	[14] 14.xxx
This is the output of this logic block and the DPT can differ from the input. The value when true or false or the result of the logic block will be sent with this object.					
159	Scene 1 input	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
This is the object to trigger the first scene. The scene number to trigger and record this first scene can be configured in the parameters.					
160	Scene 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The scene can be disabled by sending a 1 to this object.					
160	Scene 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The scene can be disabled by sending a 0 to this object.					
161	Scene 1 event 1	> On / Off	1 Bit	-WCTU-	[1] 1.xxx

µBrick Actuator Series

This is the first event for the first scene.					
161	Scene 1 event 1	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the first event for the first scene.					
161	Scene 1 event 1	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the first event for the first scene.					
161	Scene 1 event 1	> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the first event for the first scene.					
161	Scene 1 event 1	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the first event for the first scene.					
161	Scene 1 event 1	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the first event for the first scene.					
161	Scene 1 event 1	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the first event for the first scene.					
161	Scene 1 event 1	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the first event for the first scene.					
161	Scene 1 event 1	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the first event for the first scene.					
161	Scene 1 event 1	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the first event for the first scene.					
162	Scene 1 event 2	> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the second event for the first scene.					
162	Scene 1 event 2	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling

µBrick Actuator Series

This is the second event for the first scene.					
162	Scene 1 event 2	> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the second event for the first scene.					
162	Scene 1 event 2	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the second event for the first scene.					
162	Scene 1 event 2	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the second event for the first scene.					
162	Scene 1 event 2	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the second event for the first scene.					
162	Scene 1 event 2	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the second event for the first scene.					
162	Scene 1 event 2	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the second event for the first scene.					
162	Scene 1 event 2	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the second event for the first scene.					
162	Scene 1 event 2	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the second event for the first scene.					
163	Scene 1 event 3	> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the third event for the first scene.					
163	Scene 1 event 3	> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the third event for the first scene.					
163	Scene 1 event 3	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count

µBrick Actuator Series

This is the third event for the first scene.					
163	Scene 1 event 3	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the third event for the first scene.					
163	Scene 1 event 3	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the third event for the first scene.					
163	Scene 1 event 3	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the third event for the first scene.					
163	Scene 1 event 3	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the third event for the first scene.					
163	Scene 1 event 3	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the third event for the first scene.					
163	Scene 1 event 3	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the third event for the first scene.					
163	Scene 1 event 3	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the third event for the first scene.					
164	Scene 1 event 4	> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count

µBrick Actuator Series

This is the fourth event for the first scene.					
164	Scene 1 event 4	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the fourth event for the first scene.					
164	Scene 1 event 4	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fourth event for the first scene.					
165	Scene 1 event 5	> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx

µBrick Actuator Series

This is the fifth event for the first scene.					
165	Scene 1 event 5	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fifth event for the first scene.					
165	Scene 1 event 5	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the fifth event for the first scene.					
166	Scene 1 event 6	> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount

µBrick Actuator Series

This is the sixth event for the first scene.					
166	Scene 1 event 6	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the sixth event for the first scene.					
166	Scene 1 event 6	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the sixth event for the first scene.					
167	Scene 1 event 7	> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the seventh event for the first scene.					
167	Scene 1 event 7	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the seventh event for the first scene.					
167	Scene 1 event 7	< 1 byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the seventh event for the first scene.					
167	Scene 1 event 7	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the seventh event for the first scene.					
167	Scene 1 event 7	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the seventh event for the first scene.					
167	Scene 1 event 7	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the seventh event for the first scene.					
167	Scene 1 event 7	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount

µBrick Actuator Series

This is the seventh event for the first scene.					
167	Scene 1 event 7	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the seventh event for the first scene.					
167	Scene 1 event 7	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the seventh event for the first scene.					
167	Scene 1 event 7	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the seventh event for the first scene.					
168	Scene 1 event 8	> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount

µBrick Actuator Series

This is the eighth event for the first scene.					
168	Scene 1 event 8	> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the eighth event for the first scene.					
168	Scene 1 event 8	> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the eighth event for the first scene.					
259	Advanced Scene 1 input	< On / Off	1 Bit	-WC---	[1] 1.xxx
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 0..100%	1 Byte	-WC---	[5.1] DPT_Scaling
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 1 byte signed	1 Byte	-WC---	[6.10] DPT_Value_1_Count
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 1 byte unsigned	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 2 bytes unsigned	2 Bytes	-WC---	[7.1] DPT_Value_2_Ucount
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 2 bytes float	2 Bytes	-WC---	[9] 9.xxx
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 2 bytes signed	2 Bytes	-WC---	[8.1] DPT_Value_2_Count
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 4 bytes float	4 Bytes	-WC---	[14] 14.xxx
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 4 bytes signed	4 Bytes	-WC---	[13.1] DPT_Value_4_Count

µBrick Actuator Series

This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
259	Advanced Scene 1 input	< 4 bytes unsigned	4 Bytes	-WC---	[12.1] DPT_Value_4_Ucount
This is the input object to trigger a function of the advanced scene. Different values for this function can be set in the parameters like the play, record, stop and restore values.					
260	Advanced Scene 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The scene can be disable with a 1					
260	Advanced Scene 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The scene can be disable with a 0					
261	Advanced Scene 1 event 1	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount

µBrick Actuator Series

This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the first event for the first advanced scene.					
261	Advanced Scene 1 event 1	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the first event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx

µBrick Actuator Series

This is the second event for the first advanced scene.					
262	Advanced Scene 1 event 2	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the second event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the third event for the first advanced scene.					
263	Advanced Scene 1 event 3	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount

µBrick Actuator Series

This is the third event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the fourth event for the first advanced scene.					
264	Advanced Scene 1 event 4	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fourth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx

µBrick Actuator Series

This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the fifth event for the first advanced scene.					
265	Advanced Scene 1 event 5	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the fifth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 1 byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the sixth event for the first advanced scene.					
266	Advanced Scene 1 event 6	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the sixth event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the seventh event for the first advanced scene.					
267	Advanced Scene 1 event 7	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the seventh event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> On / Off	1 Bit	-WCTU-	[1] 1.xxx
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 1 byte signed	1 Byte	-WCTU-	[6.10] DPT_Value_1_Count
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 2 bytes unsigned	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 2 bytes signed	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 4 bytes signed	4 Bytes	-WCTU-	[13.1] DPT_Value_4_Count
This is the eighth event for the first advanced scene.					
268	Advanced Scene 1 event 8	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the eighth event for the first advanced scene.					
359	Timer 1 trigger	< On / Off	1 Bit	-WC---	[1] 1.xxx
This is to trigger the first timer					
359	Timer 1 trigger	< 1 byte signed	1 Byte	-WC---	[6.10] DPT_Value_1_Count
This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 1 byte scaling	1 Byte	-WC---	[5.1] DPT_Scaling
This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 1 byte unsigned	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 2 bytes unsigned	2 Bytes	-WC---	[7.1] DPT_Value_2_Ucount

µBrick Actuator Series

This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 2 bytes float	2 Bytes	-WC---	[9] 9.xxx
This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 2 bytes signed	2 Bytes	-WC---	[8.1] DPT_Value_2_Count
This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 4 bytes unsigned	4 Bytes	-WC---	[12.1] DPT_Value_4_Ucount
This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 4 bytes signed	4 Bytes	-WC---	[13.1] DPT_Value_4_Count
This is to trigger the first timer (only for delay)					
359	Timer 1 trigger	< 4 bytes float	4 Bytes	-WC---	[14] 14.xxx
This is to trigger the first timer (only for delay)					
360	Timer 1 change staircase factor	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
With this object the ON time of the timer can be changed. If the base is equal to 1 second, this object will change the time in seconds. If the base is 1 minute the value sent to the object is equal to the minutes the staircase will be ON, etc.					
361	Timer 1 warning pulse	> On / Off	1 Bit	R-CT--	[1.1] DPT_Switch
An additional object can be activated to send a warning pulse to inform that the staircase is about to expire and therefore have time to react in order to trigger it again.					
362	Timer 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The timer can be disabled by this object by sending a 0					
363	Timer 1 output	> On / Off	1 Bit	--CT--	[1.1] DPT_Switch
This is the output object of the timer.					
363	Timer 1 output	> 1 byte signed	1 Byte	--CT--	[6.10] DPT_Value_1_Count
This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 1 byte unsigned	1 Byte	--CT--	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 1 byte scaling	1 Byte	--CT--	[5.1] DPT_Scaling
This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 2 bytes float	2 Bytes	--CT--	[9] 9.xxx
This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 2 bytes unsigned	2 Bytes	--CT--	[7.1] DPT_Value_2_Ucount
This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 2 bytes signed	2 Bytes	--CT--	[8.1] DPT_Value_2_Count
This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 4 bytes signed	4 Bytes	--CT--	[13.1] DPT_Value_4_Count
This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 4 bytes unsigned	4 Bytes	--CT--	[12.1] DPT_Value_4_Ucount
This is the output object of the timer. (only for the delay function)					
363	Timer 1 output	> 4 bytes float	4 Bytes	--CT--	[14] 14.xxx
This is the output object of the timer. (only for the delay function)					
409	Setpoint 1 output regulator	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the output of the two point regulator for the first setpoint. This output will switch ON or OFF depending on the parametrized values when crossing the threshold values					
410	Setpoint 1 setpoint value/status	<> 0..100%	1 Byte	RWCT--	[5.1] DPT_Scaling
The desired setpoint value can be adjusted with this object. The same object will be used to send the current setpoint status value. This status value will be sent when changing from heat to cool and depending on the parameters when blocking and unblocking the setpoint					
410	Setpoint 1 setpoint value/status	<> 1 byte unsigned	1 Byte	RWCT--	[5.10] DPT_Value_1_Ucount
The desired setpoint value can be adjusted with this object. The same object will be used to send the current setpoint status value. This status value will be sent when changing from heat to cool and depending on the parameters when blocking and unblocking the setpoint					

µBrick Actuator Series

410	Setpoint 1 setpoint value/status	<> 2 bytes float	2 Bytes	RWCT--	[9] 9.xxx
The desired setpoint value can be adjusted with this object. The same object will be used to send the current setpoint status value. This status value will be sent when changing from heat to cool and depending on the parameters when blocking and unblocking the setpoint					
410	Setpoint 1 setpoint value/status	<> 2 bytes unsigned	2 Bytes	RWCT--	[7.1] DPT_Value_2_Ucount
The desired setpoint value can be adjusted with this object. The same object will be used to send the current setpoint status value. This status value will be sent when changing from heat to cool and depending on the parameters when blocking and unblocking the setpoint					
410	Setpoint 1 setpoint value/status	<> 4 bytes float	4 Bytes	RWCT--	[14] 14.xxx
The desired setpoint value can be adjusted with this object. The same object will be used to send the current setpoint status value. This status value will be sent when changing from heat to cool and depending on the parameters when blocking and unblocking the setpoint					
410	Setpoint 1 setpoint value/status	<> 4 bytes unsigned	4 Bytes	RWCT--	[12.1] DPT_Value_4_Ucount
The desired setpoint value can be adjusted with this object. The same object will be used to send the current setpoint status value. This status value will be sent when changing from heat to cool and depending on the parameters when blocking and unblocking the setpoint					
411	Setpoint 1 Heat / Cool	< Heat = 1 / Cool = 0	1 Bit	RWC---	[1] 1.xxx
With this object the two point regulator will change from heat to cool mode. This will cause the threshold to change from: (Lower threshold = Setpoint at Cool = 0) and (Upper threshold = Setpoint at Heat = 1)					
412	Setpoint 1 input ext. sensor value	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is the analog value which will be used as the input for the setpoint					
412	Setpoint 1 input ext. sensor value	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
This is the analog value which will be used as the input for the setpoint					
412	Setpoint 1 input ext. sensor value	< 2 bytes float	2 Bytes	RWC---	[9] 9.xxx
This is the analog value which will be used as the input for the setpoint					
412	Setpoint 1 input ext. sensor value	< 2 byte unsigned	2 Bytes	RWC---	[7.1] DPT_Value_2_Ucount
This is the analog value which will be used as the input for the setpoint					

µBrick Actuator Series

412	Setpoint 1 input ext. sensor value	< 4 bytes float	4 Bytes	RWC---	[14] 14.xxx
This is the analog value which will be used as the input for the setpoint					
412	Setpoint 1 input ext. sensor value	< 4 bytes unsigned	4 Bytes	RWC---	[12.1] DPT_Value_4_Ucount
This is the analog value which will be used as the input for the setpoint					
413	Setpoint 1 disable	< On / Off	1 Bit	RWC---	[1] 1.xxx
The setpoint can be disabled with this object					
413	Setpoint 1 disable	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
The setpoint can be disabled with this object. This can also be used to change the HVAC mode when linking this object of more than one setpoint to the same group address but with different enable values. E.g. If setpoint 1 is enabled by the value 1 and setpoint 2 by the value 2, then setpoint 1 can be the comfort mode and setpoint 2 standby mode.					
559	[A1] Switching On / Off	< On / Off	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 1/ON when configured as N.O. contact. On the other hand it will be opened when receiving a 1/ON when configured as N.C. contact.					
559	[A] Move	< 0=up/1=down	1 Bit	-WC---	[1.8] DPT_UpDown
This object is to move the blind up=0 or down=1					
560	[A1] Switching toggle/inverted	< Inverted	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
560	[A] Stop (Blind=Stop/step)	< 0=stop/step, 1=stop/step	1 Bit	-WC---	[1] 1.xxx
This is to stop/step the blind 0=stop/step up, 1=stop/step down					
560	[A1] Switching toggle/inverted	< Toggle only with 0	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
560	[A1] Switching toggle/inverted	< Toggle with 0 and 1	1 Bit	-WC---	[1.1] DPT_Switch

µBrick Actuator Series

With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
560	[A1] Switching toggle/inverted	< Toggle only with 1	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
561	[A1] Switching status	> On / Off	1 Bit	R-CT--	[1.1] DPT_Switch
This is the current status of the channel. The sending behaviour can be changed by the parameters					
561	[A] Move to position	< 0..100%	1 Byte	-WC---	[5.1] DPT_Scaling
The blind can be moved to a specific absolute position with this object.					
562	[A1] RunHour counter value	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
The runhour value of this channel will be sent to the bus. The frequency to be sent can be adjusted. It can also be set to send different values than hours, when using the advanced functions of the runhour. Please see the parameter description.					
562	[A] Move slat	< 0..100%	1 Byte	-WC---	[5.1] DPT_Scaling
This object is to move the slats to an absolute position.					
562	[A1] RunHour counter value	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
The accumulated ON time of the channel is called the runhours and it is sent by this object. The frequency and values to be sent can be changed in the application program. One can even apply different multiplying or division factors in the application.					
562	[A1] RunHour counter value	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
The accumulated ON time of the channel is called the runhours and it is sent by this object. The frequency and values to be sent can be changed in the application program. One can even apply different multiplying or division factors in the application.					
563	[A] Change upper limit	<> 0..100%	1 Byte	RWCT--	[5.1] DPT_Scaling
The blinds can have limits configured in the parameters and the upper limit can be changed by using this object. Should an invalid value (upper limit must be smaller than lower limit) be sent to this object it will be rejected and the previous value will be restored and sent to the bus.					
563	[A1] RunHour counter threshold	< Reading/writing threshold	1 Byte	RWCT--	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
563	[A1] RunHour counter threshold	< Reading threshold	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
563	[A1] RunHour counter threshold	< Reading/writing threshold	2 Bytes	RWCT--	[7.1] DPT_Value_2_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
563	[A1] RunHour counter threshold	< Reading threshold	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
563	[A1] RunHour counter threshold	< Reading/writing threshold	4 Bytes	RWCT--	[12.1] DPT_Value_4_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
563	[A1] RunHour counter threshold	< Reading threshold	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
564	[A1] RunHour counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	[1] 1.xxx
When crossing the threshold value the threshold alarm object will send an alarm message.					
564	[A] Change lower limit	<> 0..100%	1 Byte	RWCT--	[5.1] DPT_Scaling
The blinds can have limits configured in the parameters and the lower limit can be changed by using this object. Should an invalid value (upper limit must be smaller than lower limit) be sent to this object it will be rejected and the previous value will be restored and sent to the bus.					
565	[A1] RunHour counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The runhour counter can be reset by this object in order to start counting again from zero. In the parameters one can decide to reset to zero or to have the counter object should maintain and send the last value at reset					
565	[A] Status blind position	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This object sends the absolute blind status. The sending conditions can be set in the parameters.					
566	[A] Status blind lower end position	> 1 = Totally down / 0 = not	1 Bit	R-CT--	[1] 1.xxx
When reaching the lower end position this object will send a 1, for any other position this object will be 0.					

µBrick Actuator Series

566	[A1] RunHour counter value at reset	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the runhour counter at reset.					
566	[A1] RunHour counter value at reset	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the runhour counter at reset.					
566	[A1] RunHour counter value at reset	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the runhour counter at reset.					
567	[A] Status blind upper end position	> 1 = Totally up / 0 = not	1 Bit	R-CT--	[1] 1.xxx
When reaching the upper end position this object will send a 1, for any other position this object will be 0.					
567	[A1] Switching counter value	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This object sends the number of switching's, whether to count when it switches ON, OFF or both can be configured in the parameters					
567	[A1] Switching counter value	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This object sends the number of switching's, whether to count when it switches ON, OFF or both can be configured in the parameters					
567	[A1] Switching counter value	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This object sends the number of switching's, whether to count when it switches ON, OFF or both can be configured in the parameters					
568	[A] Status slat position	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This sends the status of the slat position after each movement.					
568	[A1] Switching counter threshold	< Reading/writing threshold	1 Byte	RWCT--	[5.10] DPT_Value_1_Ucount
This object is to read and write the threshold value.					
568	[A1] Switching counter threshold	< Reading threshold	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This object is to only read the threshold value.					
568	[A1] Switching counter threshold	< Reading threshold	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount

µBrick Actuator Series

This object is to only read the threshold value.					
568	[A1] Switching counter threshold	< Reading/writing threshold	2 Bytes	RWCT--	[7.1] DPT_Value_2_Ucount
This object is to read and write the threshold value.					
568	[A1] Switching counter threshold	< Reading threshold	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This object is to only read the threshold value.					
568	[A1] Switching counter threshold	< Reading/writing threshold	4 Bytes	RWCT--	[12.1] DPT_Value_4_Ucount
This object is to read and write the threshold value.					
569	[A] Preset 1 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
With a 1 this preset will be executed. 0 = No reaction					
569	[A1] Switching counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	[1] 1.xxx
When crossing the threshold value the threshold alarm object will send an alarm message.					
570	[A] Preset 2 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
With a 1 this preset will be executed. 0 = No reaction					
570	[A1] Switching counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The switching counter can be reset by this object in order to start counting again from zero. In the parameters one can decide to reset to zero or to have the counter object maintain and send the last value at reset					
571	[A] Preset 3 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
With a 1 this preset will be executed. 0 = No reaction					
571	[A1] Switching counter value at reset	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the switching counter at reset.					
571	[A1] Switching counter value at reset	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the switching counter at reset.					
571	[A1] Switching counter value at reset	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount

µBrick Actuator Series

In the parameters one can decide to activate this object and if it should store and send the last value of the switching counter at reset.					
572	[A] Preset 4 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
With a 1 this preset will be executed. 0 = No reaction					
572	[A1] Scene number	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
With this object any of the configured scenes of this channel can be triggered and/or recorded.					
573	[A1] Scene disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The scene function for this channel can be disabled by sending a 1 to this object					
573	[A1] Scene disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The scene function for this channel can be disabled by sending a 0 to this object					
573	[A] Preset 1 change move position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute movement position which will be set when calling preset 1					
574	[A1] Timer 1 trigger	< On / Off	1 Bit	-WC---	[1] 1.xxx
This is to trigger the first timer associated to the channel					
574	[A] Preset 2 change move position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute movement position which will be set when calling preset 2					
575	[A] Preset 3 change move position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute movement position which will be set when calling preset 3					
575	[A1] Timer 1 change staircase factor	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
With this object the ON time of the first timer of this channel can be changed. If the base is equal to 1 second, this object will change the time in seconds. If the base is 1 minute the value sent to the object is equal to the minutes the staircase will be ON, etc.					
576	[A1] Timer 1 warning pulse	> On / Off	1 Bit	R-CT--	[1.1] DPT_Switch
An additional object can be activated to send a warning pulse to inform that the staircase is about to expire and therefore have time to react in order to trigger it again.					
576	[A] Preset 4 change move position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling

µBrick Actuator Series

This is to change the blind absolute movement position which will be set when calling preset 4					
577	[A1] Timer 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1] 1.xxx
With this object the timer will be disabled by receiving a 0					
577	[A] Preset 1 change slat position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute slat position which will be set when calling preset 1					
578	[A1] Timer 2 trigger	< On / Off	1 Bit	-WC---	[1] 1.xxx
This is to trigger the second timer associated to the channel					
578	[A] Preset 2 change slat position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute slat position which will be set when calling preset 2					
579	[A] Preset 3 change slat position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute slat position which will be set when calling preset 3					
579	[A1] Timer 2 change staircase factor	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
With this object the ON time of the second timer of this channel can be changed. If the base is equal to 1 second, this object will change the time in seconds. If the base is 1 minute the value sent to the object is equal to the minutes the staircase will be ON, etc.					
580	[A1] Timer 2 warning pulse	> On / Off	1 Bit	R-CT--	[1.1] DPT_Switch
An additional object can be activated to send a warning pulse to inform that the staircase is about to expire and therefore have time to react in order to trigger it again.					
580	[A] Preset 4 change slat position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute slat position which will be set when calling preset 4					
581	[A] Preset 1 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The current position of the blind and/or (depending on the parameters) the slats can be saved as the new preset 1 values when sending a 1 to this object					
581	[A1] Timer 2 disable	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1] 1.xxx
The timer can be disabled by this object by sending a 0					
582	[A] Preset 2 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The current position of the blind and/or (depending on the parameters) the slats can be saved as the new preset 1 values when sending a 1 to this object					
582	[A1] Disable channel	< On / Off	1 Bit	RWCT--	[1] 1.xxx

The channel can be disabled by this object. In the parameters one can decide to disable with a 1 or a 0.					
583	[A] Preset 3 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The current position of the blind and/or (depending on the parameters) the slats can be saved as the new preset 1 values when sending a 1 to this object					
583	[A2] Switching On / Off	< On / Off	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 1/ON when configured as N.O. contact. On the other hand it will be opened when receiving a 1/ON when configured as N.C. contact.					
584	[A2] Switching toggle/inverted	< Toggle only with 1	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
584	[A] Preset 4 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The current position of the blind and/or (depending on the parameters) the slats can be saved as the new preset 1 values when sending a 1 to this object					
584	[A2] Switching toggle/inverted	< Toggle with 0 and 1	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
584	[A2] Switching toggle/inverted	< Toggle only with 0	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
584	[A2] Switching toggle/inverted	< Inverted	1 Bit	-WC---	[1.1] DPT_Switch
With this object the switching channels relay will be closed when receiving a 0/OFF when configured as N.O. contact. On the other hand it will be opened when receiving a 0/OFF when configured as N.C. contact, if so configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output. The value to do this can also be configured in the parameters					
585	[A2] Switching status	> On / Off	1 Bit	R-CT--	[1.1] DPT_Switch
This is the current status of the channel. The sending behaviour can be changed by the parameters					
585	[A] Scene number	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
With this object any of the configured scenes of this channel can be triggered and/or recorded.					

µBrick Actuator Series

586	[A] Scene disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The scene function for this channel can be disabled by sending a 1 to this object					
586	[A] Scene disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The scene function for this channel can be disabled by sending a 1 to this object					
586	[A2] RunHour counter value	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
The runhour value of this channel will be sent to the bus. The frequency to be sent can be adjusted. It can also be set to send different values than hours, when using the advanced functions of the runhour. Please see the parameter description.					
586	[A2] RunHour counter value	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
The runhour value of this channel will be sent to the bus. The frequency to be sent can be adjusted. It can also be set to send different values than hours, when using the advanced functions of the runhour. Please see the parameter description.					
586	[A2] RunHour counter value	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
The runhour value of this channel will be sent to the bus. The frequency to be sent can be adjusted. It can also be set to send different values than hours, when using the advanced functions of the runhour. Please see the parameter description.					
587	[A] Disable channel	< On / Off	1 Bit	RWCT--	[1] 1.xxx
The channel can be disabled by this object. In the parameters one can decide to disable with a 1 or a 0.					
587	[A2] RunHour counter threshold	< Reading/writing threshold	1 Byte	RWCT--	[5.10] DPT_Value_1_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
587	[A2] RunHour counter threshold	< Reading threshold	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
587	[A2] RunHour counter threshold	< Reading threshold	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
587	[A2] RunHour counter threshold	< Reading/writing threshold	2 Bytes	RWCT--	[7.1] DPT_Value_2_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
587	[A2] RunHour counter threshold	< Reading threshold	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount

µBrick Actuator Series

The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
587	[A2] RunHour counter threshold	< Reading/writing threshold	4 Bytes	RWCT--	[12.1] DPT_Value_4_Ucount
The threshold of the runhour counter can be changed by this object. When crossing the threshold value the threshold alarm object will send an alarm message.					
588	[A] Move inverted	< 1=up/0=down	1 Bit	-WC---	[1] 1.xxx
This object is to move the blind down with a 0 and up with a 1. It is very usual to send an all OFF telegram when leaving the house and mostly the clients want the blinds to go down in this case. By linking the all OFF telegram to this object instead of the normal move object the blinds will move DOWN and not UP					
588	[A2] RunHour counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	[1] 1.xxx
When crossing the threshold value the threshold alarm object will send an alarm message.					
589	[A] Disable limits / calibrate	< Disable =0 / En&calibrate =1	1 Bit	RWC---	[1] 1.xxx
With this object the limits (must be configured in the parameters) will be disabled when receiving a 0. When sending a 1 to this object the limits will be enabled and the blind will make a calibration movement.					
589	[A2] RunHour counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The runhour counter can be reset by this object in order to start counting again from zero. In the parameters one can decide to reset to zero or to have the counter object maintain and send the last value at reset					
590	[A2] RunHour counter value at reset	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the runhour counter at reset.					
590	[A2] RunHour counter value at reset	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the runhour counter at reset.					
590	[A2] RunHour counter value at reset	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the runhour counter at reset.					
591	[A2] Switching counter value	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This object sends the number of switching's, whether to count when it switches ON, OFF or both can be configured in the parameters					
591	[A2] Switching counter value	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This object sends the number of switching's, whether to count when it switches ON, OFF or both can be configured in the parameters					

µBrick Actuator Series

591	[A2] Switching counter value	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This object sends the number of switching's, whether to count when it switches ON, OFF or both can be configured in the parameters					
592	[A2] Switching counter threshold	< Reading threshold	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This object is to only read the threshold value.					
592	[A2] Switching counter threshold	< Reading/writing threshold	1 Byte	RWCT--	[5.10] DPT_Value_1_Ucount
This object is to read and write the threshold value.					
592	[A2] Switching counter threshold	< Reading/writing threshold	2 Bytes	RWCT--	[7.1] DPT_Value_2_Ucount
This object is to read and write the threshold value.					
592	[A2] Switching counter threshold	< Reading threshold	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This object is to only read the threshold value.					
592	[A2] Switching counter threshold	< Reading/writing threshold	4 Bytes	RWCT--	[12.1] DPT_Value_4_Ucount
This object is to read and write the threshold value.					
592	[A2] Switching counter threshold	< Reading threshold	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This object is to only read the threshold value.					
593	[A2] Switching counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	[1] 1.xxx
When crossing the threshold value the threshold alarm object will send an alarm message.					
594	[A2] Switching counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
The switching counter can be reset by this object in order to start counting again from zero. In the parameters one can decide to reset to zero or to have the counter object maintain and send the last value at reset					
595	[A2] Switching counter value at reset	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the switching counter at reset.					
595	[A2] Switching counter value at reset	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount

µBrick Actuator Series

In the parameters one can decide to activate this object and if it should store and send the last value of the switching counter at reset.					
595	[A2] Switching counter value at reset	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
In the parameters one can decide to activate this object and if it should store and send the last value of the switching counter at reset.					
596	[A2] Scene number	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
With this object any of the configured scenes of this channel can be triggered and/or recorded.					
597	[A2] Scene disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
The scene function for this channel can be disabled by sending a 1 to this object					
597	[A2] Scene disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
The scene function for this channel can be disabled by sending a 0 to this object					
598	[A2] Timer 1 trigger	< On / Off	1 Bit	-WC---	[1] 1.xxx
This is to trigger the first timer					
599	[A2] Timer 1 change stair-case factor	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
With this object the ON time of the first timer of this channel can be changed. If the base is equal to 1 second, this object will change the time in seconds. If the base is 1 minute the value sent to the object is equal to the minutes the staircase will be ON, etc.					
600	[A2] Timer 1 warning pulse	> On / Off	1 Bit	R-CT--	[1.1] DPT_Switch
An additional object can be activated to send a warning pulse to inform that the staircase is about to expire and therefore have time to react in order to trigger it again.					
601	[A2] Timer 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1] 1.xxx
With this object the timer will be disabled by receiving a 0					
602	[A2] Timer 2 trigger	< On / Off	1 Bit	-WC---	[1] 1.xxx
This is to trigger the second timer					
603	[A2] Timer 2 change stair-case factor	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
With this object the ON time of the second timer of this channel can be changed. If the base is equal to 1 second, this object will change the time in seconds. If the base is 1 minute the value sent to the object is equal to the minutes the staircase will be ON, etc.					
604	[A2] Timer 2 warning pulse	> On / Off	1 Bit	R-CT--	[1.1] DPT_Switch
An additional object can be activated to send a warning pulse to inform that the staircase is about to expire and therefore have time to react in order to trigger it again.					

µBrick Actuator Series

605	[A2] Timer 2 disable	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1] 1.xxx
With this object the timer will be disabled by receiving a 0					
606	[A2] Disable channel	< On / Off	1 Bit	RWCT--	[1] 1.xxx
The channel can be disabled by this object. In the parameters one can decide to disable with a 1 or a 0.					
703	[In1] Disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx
This is to disable the first input by sending a 1 to this object.					
703	[In1] Disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
This is to disable the first input by sending a 0 to this object.					
704	[In1] Switching short	> On / Off	1 Bit	RWCT--	[1.1] DPT_Switch
This is the action to be sent to the bus when pressing the button short. (The time for long operation can be configured in the parameters)					
704	[In1] Switching short	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This is the action to be sent to the bus when pressing the button short. (The time for long operation can be configured in the parameters)					
704	[In1] Switching short	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the action to be sent to the bus when pressing the button short. (The time for long operation can be configured in the parameters)					
704	[In1] Switching short	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the action to be sent to the bus when pressing the button short. (The time for long operation can be configured in the parameters)					
704	[In1] Switching short	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This is the action to be sent to the bus when pressing the button short. (The time for long operation can be configured in the parameters)					
704	[In1] Switching short	> 4 bytes float	4 Bytes	R-CT--	[14] 14.xxx
This is the action to be sent to the bus when pressing the button short. (The time for long operation can be configured in the parameters)					
705	[In1] Switching long	> On / Off	1 Bit	RWCT--	[1.1] DPT_Switch
This is the action to be sent to the bus when pressing the button long. (The time for long operation can be configured in the parameters)					
705	[In1] Switching long	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling

µBrick Actuator Series

This is the action to be sent to the bus when pressing the button long. (The time for long operation can be configured in the parameters)					
705	[In1] Switching long	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the action to be sent to the bus when pressing the button long. (The time for long operation can be configured in the parameters)					
705	[In1] Switching long	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the action to be sent to the bus when pressing the button long. (The time for long operation can be configured in the parameters)					
705	[In1] Switching long	> 4 bytes float	4 Bytes	R-CT--	[14] 14.xxx
This is the action to be sent to the bus when pressing the button long. (The time for long operation can be configured in the parameters)					
705	[In1] Switching long	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This is the action to be sent to the bus when pressing the button long. (The time for long operation can be configured in the parameters)					
706	[In1] Multiple op. 1 pulse	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the first multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
706	[In1] Multiple op. 1 pulse	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This is the first multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
706	[In1] Multiple op. 1 pulse	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the first multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
706	[In1] Multiple op. 1 pulse	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the first multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
707	[In1] Multiple op. 2 pulses	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the second multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
707	[In1] Multiple op. 2 pulses	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This is the second multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
707	[In1] Multiple op. 2 pulses	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This is the second multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
707	[In1] Multiple op. 2 pulses	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the second multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
708	[In1] Multiple op. 3 pulses	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the third multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
708	[In1] Multiple op. 3 pulses	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This is the third multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
708	[In1] Multiple op. 3 pulses	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the third multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
708	[In1] Multiple op. 3 pulses	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the third multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
709	[In1] Multiple op. 4 pulses	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the fourth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
709	[In1] Multiple op. 4 pulses	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This is the fourth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
709	[In1] Multiple op. 4 pulses	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the fourth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
709	[In1] Multiple op. 4 pulses	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the fourth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
710	[In1] Multiple op. 5 pulses	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the fifth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
710	[In1] Multiple op. 5 pulses	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the fifth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					

µBrick Actuator Series

710	[In1] Multiple op. 5 pulses	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This is the fifth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
710	[In1] Multiple op. 5 pulses	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
This is the fifth multiple operation object. The number of pulses to trigger this object can be changed in the parameters. Also the time between pulses and the value to be sent can be changed in the parameters.					
711	[In1] Multiple op. long	> On / Off	1 Bit	R-CT--	[1] 1.xxx
It is also possible to configure for the multiple operation a time for long operation. If the button is pressed longer than this time this object will send the parametrized value					
711	[In1] Multiple op. long	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
It is also possible to configure for the multiple operation a time for long operation. If the button is pressed longer than this time this object will send the parametrized value					
711	[In1] Multiple op. long	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
It is also possible to configure for the multiple operation a time for long operation. If the button is pressed longer than this time this object will send the parametrized value					
711	[In1] Multiple op. long	> 2 bytes float	2 Bytes	R-CT--	[9] 9.xxx
It is also possible to configure for the multiple operation a time for long operation. If the button is pressed longer than this time this object will send the parametrized value					
712	[In1] Flashing	> On / Off	1 Bit	R-CT--	[1] 1.xxx
This is the object to send the flashing sequence to the bus. The ON and OFF time can individually be adjusted in the parameters.					
713	[In1] Dimming on/off	> On / Off	1 Bit	-WCT--	[1.1] DPT_Switch
This is the ON/OFF telegram generated when pressing the button short if the input is configured to have a dimming function.					
714	[In1] Dimming +/-	> 4 bits relative dimming	4 Bit	-WCT--	[3.7] DPT_Control_Dimming
This is the 4 bit relative dimming telegram generated when pressing the button long if the input is configured to have a dimming function. The step size and whether or not a stop telegram must be set can be configured in the parameters.					
715	[In1] Blind move	> Up = 0 / Down = 1	1 Bit	-WCT--	[1.8] DPT_UpDown
This object is to move the blinds up or down according to the KNX DPT 1.008 with a long press of the button					
716	[In1] Blind stop/step	> Step Up = 0 / Step Down = 1	1 Bit	-WCT--	[1] 1.xxx
This object is to move the slats up or down or to stop the blind according to the KNX DPT 1.007 with a short press of the button					
717	[In1] Scene	> Sc1 (0=Play 128=Rec)... Sc64	1 Byte	--CT--	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This sends the scene number to the bus with a short press of the button and sends a record telegram with a long press of the button.					
718	[In1] Sequence output 1	> On / Off	1 Bit	-WCT--	[1] 1.xxx
This is the first (out of max. 4) sequence output object of the first input and will send a value to the bus depending on the parametrized value. Depending on the type of sequence the output objects will sequentially switch ON or OFF (increment/decrement)					
718	[In1] Sequence output 1	> 1 byte unsigned	1 Byte	-WCT--	[5.10] DPT_Value_1_Ucount
This is the first (out of max. 4) sequence output object of the first input and will send a value to the bus depending on the parametrized value. Depending on the type of sequence the output objects will sequentially switch ON or OFF (increment/decrement)					
718	[In1] Sequence output 1	> 0..100%	1 Byte	-WCT--	[5.1] DPT_Scaling
This is the first (out of max. 4) sequence output object of the first input and will send a value to the bus depending on the parametrized value. Depending on the type of sequence the output objects will sequentially switch ON or OFF (increment/decrement)					
718	[In1] Sequence output 1	> 2 bytes float	2 Bytes	-WCT--	[9] 9.xxx
This is the first (out of max. 4) sequence output object of the first input and will send a value to the bus depending on the parametrized value. Depending on the type of sequence the output objects will sequentially switch ON or OFF (increment/decrement)					
719	[In1] Sequence output 2	> On / Off	1 Bit	-WCT--	[1] 1.xxx
This is the second (out of max. 4) sequence output object of the first input and will send a value to the bus depending on the parametrized value. Depending on the type of sequence the output objects will sequentially switch ON or OFF (increment/decrement)					
720	[In1] Sequence output 3	> On / Off	1 Bit	-WCT--	[1] 1.xxx
This is the third (out of max. 4) sequence output object of the first input and will send a value to the bus depending on the parametrized value. Depending on the type of sequence the output objects will sequentially switch ON or OFF (increment/decrement)					
721	[In1] Sequence output 4	> On / Off	1 Bit	-WCT--	[1] 1.xxx
This is the fourth (out of max. 4) sequence output object of the first input and will send a value to the bus depending on the parametrized value. Depending on the type of sequence the output objects will sequentially switch ON or OFF (increment/decrement)					
722	[In1] Sequence trigger	< On = Trigger / Off = Nothing	1 Bit	-WC---	[1] 1.xxx
The sequence can be triggered from the bus with this object. This will do the same as if the input button is pressed.					
723	[In1] Sequence trigger inverted	< On = Trigger inv. / Off = No	1 Bit	-WC---	[1] 1.xxx
The sequence can be inverted from the bus with this trigger object.					
724	[In1] Counter	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the output object to send the current counter value of this input to the bus. The counter can increase its value on rising and/or falling edge.					

µBrick Actuator Series

724	[In1] Counter	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This is the output object to send the current counter value of this input to the bus. The counter can increase its value on rising and/or falling edge.					
724	[In1] Counter	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This is the output object to send the current counter value of this input to the bus. The counter can increase its value on rising and/or falling edge.					
725	[In1] Counter threshold	< Reading/writing threshold	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
This object is to read/write the threshold value of the counter					
725	[In1] Counter threshold	< Reading threshold	1 Byte	R-C---	[5.10] DPT_Value_1_Ucount
This object is to only read the threshold value of the counter					
725	[In1] Counter threshold	< Reading/writing threshold	2 Bytes	RWC---	[7.1] DPT_Value_2_Ucount
This object is to read/write the threshold value of the counter					
725	[In1] Counter threshold	< Reading threshold	2 Bytes	R-C---	[7.1] DPT_Value_2_Ucount
This object is to only read the threshold value of the counter					
725	[In1] Counter threshold	< Reading/writing threshold	4 Bytes	RWC---	[12.1] DPT_Value_4_Ucount
This object is to read/write the threshold value of the counter					
725	[In1] Counter threshold	< Reading threshold	4 Bytes	R-C---	[12.1] DPT_Value_4_Ucount
This object is to only read the threshold value of the counter					
726	[In1] Counter alarm	> 1=Alarm, 0=No, < 0=Reset	1 Bit	RWCT--	[1] 1.xxx
This sends an alarm message if the threshold of the counter has been reached.					
727	[In1] Counter reset	< On = Reset / Off = Nothing	1 Bit	-WC---	[1] 1.xxx
With this object the counter can be reset. If the threshold has been reached the 1 bit “Counter alarm” object will be equal to “1” indicating alarm. This alarm object will reset to zero when receiving a “1” on this “[In1] Counter reset” object, but it will not be sent to the bus.					
s	[In1] Counter last value	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount

µBrick Actuator Series

This is the last value of the counter at reset					
728	[In1] Counter last value	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This is the last value of the counter at reset					
728	[In1] Counter last value	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This is the last value of the counter at reset					
729	[In1] Counter trigger input	< On = Trigger / Off = Trigger	1 Bit	-WC---	[1] 1.xxx
The counter can also be triggered with a telegram from the bus. This will trigger the counter when receiving OFF and ON telegrams					
729	[In1] Counter trigger input	< On = Nothing / Off = Trigger	1 Bit	-WC---	[1] 1.xxx
The counter can also be triggered with a telegram from the bus. This will trigger the counter when receiving OFF telegrams					
729	[In1] Counter trigger input	< On = Trigger / Off = Nothing	1 Bit	-WC---	[1] 1.xxx
The counter can also be triggered with a telegram from the bus. This will trigger the counter when receiving ON telegrams					
730	[In1] Counter additional count.	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This object counts the same input signal, but it can have different trigger parameters than the main counter. E.g. This additional counter can be used to get daily values by resetting it every 24 hours.					
730	[In1] Counter additional count.	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This object counts the same input signal, but it can have different trigger parameters, than the main counter. E.g. This additional counter can be used to get daily values by resetting it every 24 hours.					
730	[In1] Counter additional count.	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This object counts the same input signal, but it can have different trigger parameters, than the main counter. E.g. This additional counter can be used to get daily values by resetting it every 24 hours.					
731	[In1] Counter additional count. reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1] 1.xxx
This is to reset the additional counter with a 1					
732	[In1] Counter additional count. last value	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the object to store the last value of the additional counter at reset.					

µBrick Actuator Series

732	[In1] Counter additional count. last value	> 2 bytes unsigned	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This is the object to store the last value of the additional counter at reset.					
732	[In1] Counter additional count. last value	> 4 bytes unsigned	4 Bytes	R-CT--	[12.1] DPT_Value_4_Ucount
This is the object to store the last value of the additional counter at reset.					
733	[In1] MD lighting output	> On / Off	1 Bit	--CT--	[1.1] DPT_Switch
This object will send the parametrized lighting output value when the movement detector detects a movement.					
733	[In1] MD lighting output	> 1 byte unsigned	1 Byte	--CT--	[5.10] DPT_Value_1_Ucount
This object will send the parametrized lighting output value when the movement detector detects a movement.					
733	[In1] MD lighting output	> 0..100%	1 Byte	--CT--	[5.1] DPT_Scaling
This object will send the parametrized lighting output value when the movement detector detects a movement.					
733	[In1] MD lighting output	> 2 bytes float	2 Bytes	--CT--	[9] 9.xxx
This object will send the parametrized lighting output value when the movement detector detects a movement.					
733	[In1] MD lighting output	> 4 bytes float	4 Bytes	--CT--	[14] 14.xxx
This object will send the parametrized lighting output value when the movement detector detects a movement.					
733	[In1] MD lighting output	> 4 bytes unsigned	4 Bytes	--CT--	[12.1] DPT_Value_4_Ucount
This object will send the parametrized lighting output value when the movement detector detects a movement.					
734	[In1] MD lighting LUX input	< 2 bytes float	2 Bytes	RWC---	[9.4] DPT_Value_Lux
When configured to switch the light ON or OFF depending on the brightness by an additional object, this object is used to receive the brightness value from the bus.					
735	[In1] MD lighting disable 1	< Disable = 1 / Enable = 0	1 Bit	-WC---	[1] 1.xxx
This is the first lighting disable input object and will disable the movement detector when receiving a 1. This object only is an input object and does not reflect the status whether or not it is blocked, for that there is an additional status object.					
735	[In1] MD lighting disable 1	< Disable = 0 / Enable = 1	1 Bit	-WC---	[1] 1.xxx
This is the first lighting disable input object and will disable the movement detector when receiving a 0. This object only is an input object and does not reflect the status whether or not it is blocked, for that there is an additional status object.					

µBrick Actuator Series

736	[In1] MD lighting disable 2	< Disable = 0 / Enable = 1	1 Bit	-WC---	[1] 1.xxx
This is the second lighting disable input object and will disable the movement detector when receiving a 1. This object only is an input object and does not reflect the status whether or not it is blocked, for that there is an additional status object.					
736	[In1] MD lighting disable 2	< Disable = 1 / Enable = 0	1 Bit	-WC---	[1] 1.xxx
This is the second lighting disable input object and will disable the movement detector when receiving a 0. This object only is an input object and does not reflect the status whether or not it is blocked, for that there is an additional status object.					
737	[In1] MD lighting status	> Disable = 1 / Enable = 0	1 Bit	R-CT--	[1] 1.xxx
This is the status telegram to indicate if the lighting channel of the detector is blocked or not. The status value will be 1 when the channel is disabled and a 0 when enabled					
738	[In1] MD HVAC output	> On / Off	1 Bit	--CT--	[1.1] DPT_Switch
This is the HVAC output object for the movement detector and will send the parametrized value to the bus depending of the settings in the parameters. By default it will not immediately send a telegram on detection, but only after detecting for a set time.					
738	[In1] MD HVAC output	> 0..100%	1 Byte	--CT--	[5.1] DPT_Scaling
This is the HVAC output object for the movement detector and will send the parametrized value to the bus depending of the settings in the parameters. By default it will not immediately send a telegram on detection, but only after detecting for a set time.					
738	[In1] MD HVAC output	> 1 byte unsigned	1 Byte	--CT--	[5.10] DPT_Value_1_Ucount
This is the HVAC output object for the movement detector and will send the parametrized value to the bus depending of the settings in the parameters. By default it will not immediately send a telegram on detection, but only after detecting for a set time.					
738	[In1] MD HVAC output	> 2 bytes float	2 Bytes	--CT--	[9] 9.xxx
This is the HVAC output object for the movement detector and will send the parametrized value to the bus depending of the settings in the parameters. By default it will not immediately send a telegram on detection, but only after detecting for a set time.					
738	[In1] MD HVAC output	> 4 bytes float	4 Bytes	--CT--	[14] 14.xxx
This is the HVAC output object for the movement detector and will send the parametrized value to the bus depending of the settings in the parameters. By default it will not immediately send a telegram on detection, but only after detecting for a set time.					
738	[In1] MD HVAC output	> 4 bytes unsigned	4 Bytes	--CT--	[12.1] DPT_Value_4_Ucount
This is the HVAC output object for the movement detector and will send the parametrized value to the bus depending of the settings in the parameters. By default it will not immediately send a telegram on detection, but only after detecting for a set time.					
739	[In1] MD HVAC disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1] 1.xxx

µBrick Actuator Series

This will disable the HVAC channel when receiving a 1					
739	[In1] MD HVAC disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1] 1.xxx
This will disable the HVAC channel when receiving a 0					
740	[In1] Temperature sensor value	> 2 bytes float	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
This is the measured temperature sensor value which will be sent to the bus depending on the parameter settings.					
740	[In1] Temperature sensor value	> 4 bytes float	4 Bytes	R-CT--	[14.68] DPT_Value_Common_Temperature
This is the measured temperature sensor value which will be sent to the bus depending on the parameter settings.					
741	[In1] Temperature external value	< 2 bytes float	2 Bytes	RWC---	[9.1] DPT_Value_Temp
The temperature can be a weighted mixture between two values, the sensor value and this object value. The proportion of each can be changed in the parameters.					
741	[In1] Temperature external value	< 4 bytes float	4 Bytes	RWC---	[14.68] DPT_Value_Common_Temperature
The temperature can be a weighted mixture between two values, the sensor value and this object value. The proportion of each can be changed in the parameters.					
742	[In1] Temperature weighted value	> 2 bytes float	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
This object sends the weighted mixture between two values, the sensor value and the "[In1] Temperature external value" object value. The proportion of each can be changed in the parameters.					
742	[In1] Temperature weighted value	> 4 bytes float	4 Bytes	R-CT--	[14.68] DPT_Value_Common_Temperature
This object sends the weighted mixture between two values, the sensor value and the "[In1] Temperature external value" object value. The proportion of each can be changed in the parameters.					
743	[In1] Temperature source supervision	> On = Error src. 1 / Off = OK	1 Bit	R-CT--	[1] 1.xxx
It is possible to supervise both the first and the second source. This object will send a 1 if there is an error in source 1					
743	[In1] Temperature source supervision	> On=Error src1 or 2 / Off=OK	1 Bit	R-CT--	[1] 1.xxx
It is possible to supervise both the first and the second source. This object will send a 1 if there is an error in any of the sources					
743	[In1] Temperature source supervision	> On = Error src. 2 / Off = OK	1 Bit	R-CT--	[1] 1.xxx

µBrick Actuator Series

It is possible to supervise both the first and the second source. This object will send a 1 if there is an error in source 2					
744	[In1] Alarm short circuit	> Alarm = Toggle, No alarm = X	1 Bit	R-CT--	[1] 1.xxx
This object sends a toggle telegram when the input detects a short circuit between the “C” common terminal and the input terminal. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
744	[In1] Alarm short circuit	> No alarm = Toggle, Alarm = X	1 Bit	R-CT--	[1] 1.xxx
This object sends a toggle telegram when the input detects a short circuit between the “C” common terminal and the input terminal. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
744	[In1] Alarm short circuit	> Alarm = 0, No alarm = 1	1 Bit	R-CT--	[1] 1.xxx
This object sends an OFF telegram when the input detects a short circuit between the “C” common terminal and the input terminal and an ON when the short circuit opens again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
744	[In1] Alarm short circuit	> Alarm = 1, No alarm = 0	1 Bit	R-CT--	[1] 1.xxx
This object sends an ON telegram when the input detects a short circuit between the “C” common terminal and the input terminal and an OFF when the short circuit opens again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
745	[In1] Alarm open circuit	> Alarm = 1, No alarm = 0	1 Bit	R-CT--	[1] 1.xxx
This object sends an ON telegram when the input detects an open circuit between the “C” common terminal and the input terminal and an OFF when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
745	[In1] Alarm open circuit	> Alarm = 0, No alarm = 1	1 Bit	R-CT--	[1] 1.xxx
This object sends an OFF telegram when the input detects an open circuit between the “C” common terminal and the input terminal and an ON when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
745	[In1] Alarm open circuit	> No alarm = Toggle, Alarm = X	1 Bit	R-CT--	[1] 1.xxx
This object does nothing when the input detects an open circuit between the “C” common terminal and the input terminal and toggles when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
745	[In1] Alarm open circuit	> Alarm = Toggle, No alarm = X	1 Bit	R-CT--	[1] 1.xxx
This object toggles when the input detects an open circuit between the “C” common terminal and the input terminal and does nothing when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
746	[In1] Alarm open / short circuit	> Alarm = 0, No alarm = 1	1 Bit	R-CT--	[1] 1.xxx

µBrick Actuator Series

This object sends an OFF telegram when the input detects an open circuit or a closed circuit between the “C” common terminal and the input terminal and an ON when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
746	[In1] Alarm open / short circuit	> Alarm = Toggle, No alarm = X	1 Bit	R-CT--	[1] 1.xxx
This object does nothing when the input detects an open circuit or a closed circuit between the “C” common terminal and the input terminal and toggles when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
746	[In1] Alarm open / short circuit	> Alarm = 1, No alarm = 0	1 Bit	R-CT--	[1] 1.xxx
This object sends an ON telegram when the input detects an open circuit or a closed circuit between the “C” common terminal and the input terminal and an OFF when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
746	[In1] Alarm open / short circuit	> No alarm = Toggle, Alarm = X	1 Bit	R-CT--	[1] 1.xxx
This object does nothing when the input detects an open circuit or a closed circuit between the “C” common terminal and the input terminal and toggles when the open circuit closes again. To use this function the 2,7k Ohm resistor (included in the box) must be connected to the end of the input line.					
747	[In1] Monitor input ACK	< Ack. with 0	1 Bit	RWC---	[1] 1.xxx
This is to acknowledge the input with a 0					
747	[In1] Monitor input ACK	< Ack. with 1	1 Bit	RWC---	[1] 1.xxx
This is to acknowledge the input with a 1					
961	Alarm 1 status	> ON = Alarm, OFF = No alarm	1 Bit	R-CT--	[1] 1.xxx
This is the alarm 1 status object and it will indicate with a 1 if there is an alarm and send a 0 if there is no alarm					

µBrick Actuator Series

3. Parameter page: General Settings

Parameter	Settings
DEVICE NAME	**µBrick**
Here a personalized name for each device can be entered. E.g. **µBrick living room**	
Inputs	No **Yes**
Use this parameter to activate or deactivate all input parameters and their objects.	
Outputs	No **Yes**
Use this parameter to activate or deactivate all outputs parameters and their objects. The outputs of the actuator are by default activated. Nevertheless, this device can also be used as an advanced controller module for logic functions, timers, etc. In this case, you can deactivate the outputs totally and completely hide all their options and objects by selecting “No”.	
ADVANCED FUNCTIONS	
All advanced features of the µBrick actuator can be activated or hidden as desired. It also serves as useful overview of all the functions available. These functions are totally channel-independent. You could even deactivate the inputs/outputs totally, thus converting the device into a pure controller module	
Alarms	**No** Yes
Use this parameter to activate or deactivate all alarm parameters and their objects.	
Logics	**No** Yes
Use this parameter to activate or deactivate all logic parameters and their objects.	
Scene controller	**No** Yes
Use this parameter to activate or deactivate all scene controller parameters and their objects.	
Advanced scene controller	**No** Yes
Use this parameter to activate or deactivate all advanced scene controller parameters and their objects.	
Timers	**No** Yes
Use this parameter to activate or deactivate all timer parameters and their objects.	
Setpoints	**No** Yes
Use this parameter to activate or deactivate all setpoint parameters and their objects.	
Internal variables	**No** Yes
Use this parameter to activate or deactivate all parameters for the internal variables.	
Overwrite end-user parameter values at download	No **Yes** Custom
By selecting “no” the end-user parameters will not be overwritten when downloading the application with the ETS. When selecting Custom the “ENDUSER PARAMETERS” tab will be activated in which almost each end-user parameter can be individually selected whether to overwrite or not.	
Central sending object for monitoring device	**No** Yes
Use this parameter to activate or deactivate the “Central cyclic telegram for monitoring” object. This object will send a cyclic ON telegram to the bus in order to supervise the device.	
Behaviour at bus recovery	**No** Yes
Use this parameter to activate or deactivate the behaviour at bus recovery.	

4.A. Parameter page: BINARY INPUTS

There are 6 inputs which can be configured to receive binary (push buttons, window contacts, water leakage sensor…) and analog signals (movement detector, temperature sensor and monitored input…)

Parameter	Settings
Input 1	No function **Binary input** Movement detector Temperature sensor

Parameter page: InX Binary input

Parameter	Settings
Type of input	**Switching / value** Dimming Shutter KNX Scene Multiple operations Flashing Sequence Counter

4.A.1 Parameter page: InX Binary input / Switching / **value**

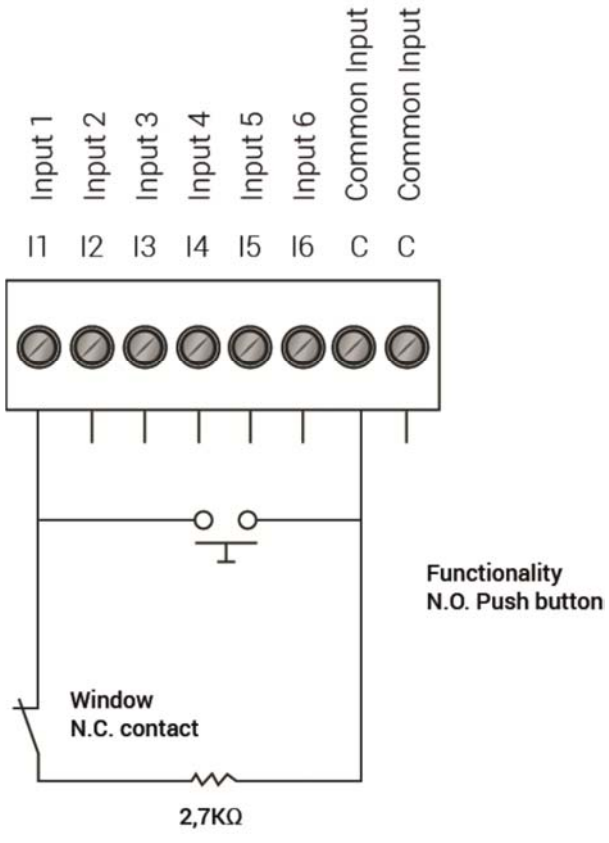
Parameter	Settings
Type of input	**Switching / value**
To send values to the bus depending of the next parameters.	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms
This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Monitoring / Doubling inputs	**No** Yes
By selecting yes the inputs can be supervised in order to generate an alarm if the input connexion has been tampered with. To do this a 2,7k Ohm resistor must be connected to the end of the input line. **Monitoring input (Open and/or Short circuit alarm detection):** The same input used with a switching / value function can be used to connect an alarm contact (like a window contact, tamper contact, etc.) with a 2,7 k ohm end of line resistor. It supervises this line and can send an alarm telegram when detecting either an alarm. It is the only device which can distinguish between short and open circuit alarms with three alarm objects. One object for the short circuit alarm, another for the open circuit alarm, and a third one which is a logic OR between the two latter. Also with or without ACK. **Doubling function:** Use monitoring input to double the binary input function (normal binary input functionality + toggle function in monitoring alarm). With 6 inputs, the device expands the inputs to be effectively used as up to 12 binary inputs.	

4.A.1.1 Parameter page: InX Binary input / Switching / **value / Monitoring input**

Parameter	Settings
Type of monitoring input / Connected contacts	**Open Circuit Alarm (N.C. contact)** Short Circuit Alarm (N.O. contact) Both (N.C. & N.O. Alarm contact)

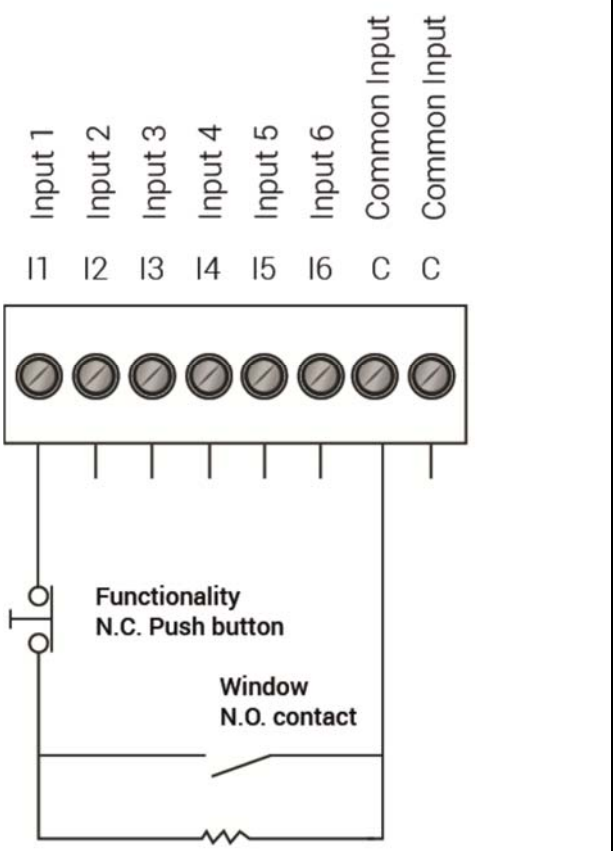
µBrick Actuator Series

MONITORING INPUT
TYPE 1: Open Circuit Alarm (N.C. contact)



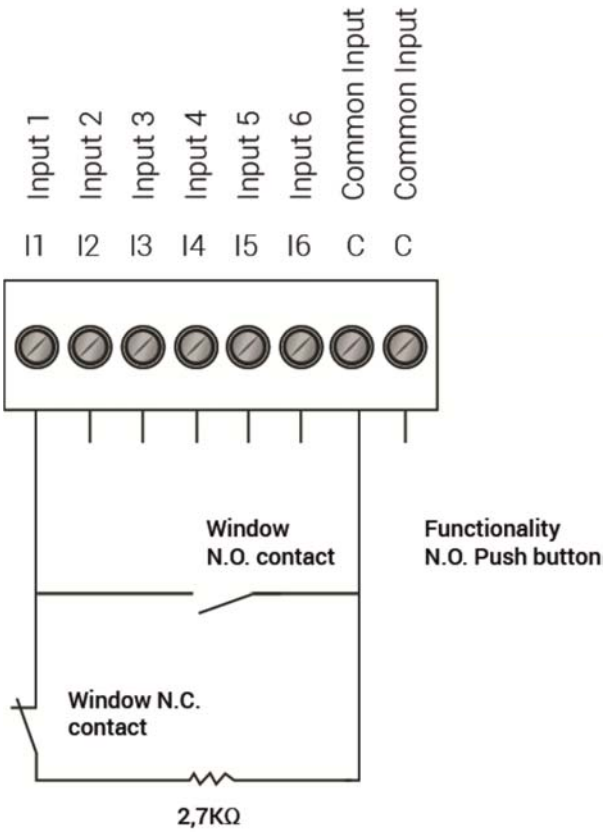
Up to 2 different funtions per input

MONITORING INPUT
TYPE 2: Short Circuit Alarm (N.O. contact)



µBrick Actuator Series

MONITORING INPUT
TYPE 3: Both (N.C. & N.O. Alarm contact)



Parameter	Settings
Type of monitoring input / Connected contacts	**Open Circuit Alarm (N.C. contact)** Short Circuit Alarm (N.O. contact) Both (N.C. & N.O. Alarm contact)
There are three possible configurations for the monitoring input.	

Type 1 - **Open circuit alarm (N.C. contact):** In this configuration the alarm contact must be a normally closed contact in series with the 2,7 k Ohm resistor between the "C" common terminal and the input. As soon as the circuit is opened (by opening the contact or by cutting the wire) it detects this and sends an alarm telegram with the "[InX] Monitor in. Alarm open circuit" object.
The push button for the switching function must have a normally opened contact and it must be connected in parallel with this contact. The push button must be closer to the input than the contact (before the N.C. alarm contact). Should the button be pressed and at the same time (while the button is pressed) an alarm comes (window is opened) the alarm will not be detected. But when releasing the button the alarm will be detected (given the alarm is still there – window is still open) and sent to the bus.

Open circuit alarm	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X Alarm = X, No alarm = Toggle
When selecting any of the alarm functions the "[InX] Monitor in. Alarm open circuit" object will be activated. With the above options one can select what value (nothing, Off, On, Toggle) should be sent with an open circuit alarm and also what value (nothing, Off, On, Toggle) when the alarm goes away.	

Parameter	Settings
Type of monitoring input / Connected contacts	**Open Circuit Alarm (N.C. contact)** Short Circuit Alarm (N.O. contact) Both (N.C. & N.O. Alarm contact)
There are three possible configurations for the monitoring input. **Type 2 - Short Circuit Alarm (N.O. contact):** In this configuration the alarm contact must be a normally opened contact in parallel with the 2,7 k Ohm resistor between the "C" common terminal and the input. As soon as the circuit is	

closed (by closing the contact or by shorting the wire) it detects this and sends an alarm telegram with the "[InX] Monitor in. Alarm short circuit" object. The push button for the switching function must have a normally closed contact and it must be connected in series with this contact. The push button must be closer to the input than the contact (before the N.O. alarm contact). Should the button be pressed and at the same time (while the button is pressed) an alarm comes (window is opened) the alarm will not be detected. But when releasing the button the alarm will be detected (given the alarm is still there – window is still open) and sent to the bus.	
Short circuit alarm	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X Alarm = X, No alarm = Toggle
When selecting any of the alarm functions the "[InX] Monitor in. Alarm short circuit" object will be activated. With the above options one can select what value (nothing, Off, On, Toggle) should be sent with an short circuit alarm and also what value (nothing, Off, On, Toggle) when the alarm goes away.	

Parameter	Settings
Type of monitoring input / Connected contacts	**Open Circuit Alarm (N.C. contact)** Short Circuit Alarm (N.O. contact) Both (N.C. & N.O. Alarm contact)

There are three possible configurations for the monitoring input.

Type 3 - Both (N.C. & N.O. Alarm contact): In this configuration there can be two different alarm contacts. One of the alarm contacts must be a normally closed contact in series with the 2,7 k Ohm resistor between the "C" common terminal and the input. As soon as this circuit is opened (by opening the contact or by cutting the wire) it detects this and sends an alarm telegram with the "[InX] Monitor in. Alarm open circuit" object (if activated). And the other alarm contact must be a normally opened contact in parallel with the 2,7 k Ohm resistor between the "C" common terminal and the input. As soon as this circuit is closed (by closing the contact or by shorting the wire) it detects this and sends an alarm telegram with the "[InX] Monitor in. Alarm short circuit" object (if activated). By default the additional "[InX] Monitor in. Alarm open / short circuit" object is activated and sends an alarm telegram if either a short circuit or an open circuit is detected. This is the most secure method because it detects any kind of tampering with the line. It detects when someone cuts the wire or tries to shorts circuit the contact.
No pushbutton should be used in this configuration. The binary function will be associated only to the N.O. contact.

Open circuit alarm	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X Alarm = X, No alarm = Toggle
When selecting any of the alarm functions the "[InX] Monitor in. Alarm open circuit" object will be activated. With the above options one can select what value (nothing, Off, On, Toggle) should be sent with an open circuit alarm and also what value (nothing, Off, On, Toggle) when the alarm goes away.	
Short circuit alarm	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X Alarm = X, No alarm = Toggle
When selecting any of the alarm functions the "[InX] Monitor in. Alarm short circuit" object will be activated. With the above options one can select what value (nothing, Off, On, Toggle) should be sent with an short circuit alarm and also what value (nothing, Off, On, Toggle) when the alarm goes away.	
Open / Short circuit alarm (N.C. & N.O. contact)	No **Alarm = 1, No alarm = 0** Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X Alarm = X, No alarm = Toggle
When selecting any of the alarm functions the "[InX] Monitor in. Alarm open / short circuit" object will be activated. With the above options one can select what value (nothing, Off, On, Toggle) should be sent with any alarm and also what value (nothing, Off, On, Toggle) when the alarm goes away.	

Parameter	Settings
Cyclic sending for all alarm objects	**No** Alarm

	No alarm Alarm & No alarm
The alarm objects can be cyclic sent on an alarm, or with no alarm, or always (both with and without alarm)	
Acknowledge needed	Ack. with 0 Ack. with 1 **No**
This is to activate the acknowledge function. The alarm can only be acknowledged if the input is not in the alarm state. One can acknowledge either with a 1 or a 0 depending on the above selection. Only after the acknowledge the alarm will go away.	
Arm / Disarm monitoring input	**No** Arm = 1 / Disarm = 0 Arm = 0 / Disarm = 1
The monitor input can be deactivated and activated independently from the binary function. With the above option one can arm (activate) the monitoring input with a 1 or a 0 (depending on the above selection) and disarm (deactivate) the monitoring input with a 1 or a 0 (depending on the above selection)	

4.A.1.1 Parameter page: Switching / value

Parameter	Settings
Type of switching function	**Short operation** Short + Long operation Short + Long operation advanced
This parameter is to select the way the input will be operated. With Short operation one can have different events for rising and falling edge. Whereas with the other two selections the events for short and long operation can be selected.	

4.A.1.1.1 Parameter page: Switching / value / Short **operation**

Parameter	Settings
Type of switching function	**Short operation**
Here one can have different events for “Event on closing the contact” rising edge and “Event on opening the contact” falling edge.	

Datapoint type short operation object	**1 bit** 1 byte scaling 1 byte unsigned 2 bytes float 4 bytes unsigned 4 bytes float
Here the Datapoint type for the short operation object can be selected.	
Event on closing the contact	**Toggle** On Off No function
A telegram with one of the above options (if DPT=1 bit where Toggle = opposite to the objects value) as its useful data will be sent when closing the contact. (rising edge) By changing the DPT the value to be sent can be introduced in an input field and the possible range depends on the DPT selection. For 2 byte float values the introduced value will be multiplied by 0.1 in order to send decimal values.	
Event on opening the contact	Toggle On Off **No function**
A telegram with one of the above options (if DPT=1 bit where Toggle = opposite to the objects value) as its useful data will be sent when opening the contact. (falling edge) By changing the DPT the value to be sent can be introduced in an input field and the possible range depends on the DPT selection. For 2 byte float values the introduced value will be multiplied by 0.1 in order to send decimal values.	
Delay of telegram	**No** At closing At opening Both
The telegram can be delayed from 1 to 255s for any of the above options.	
Cyclic sending for	**No** Closing Opening Both

The telegram can be repeated cyclically for any of the above options. Whether or not the cyclic sending can be stopped with by enabling and/or disabling the input can also be configured.	
Send input status after bus recovery	**No** Yes
The last input status can be saved on bus voltage failure and will be sent to the bus (the initial sending delay can be adjusted in the general setting tab) on bus voltage recovery if yes is selected.	

4.A.1.1.2 Parameter page: Switching / value / Short + Long operation
4.A.1.1.3 Parameter page: Switching / value / Short + Long operation advanced

Parameter	Settings
Type of switching function	Short + Long operation advanced
Attention! Advanced = event for short + event for long + event for opening after long	
SHORT OPERATION	No **Yes**
This parameter is to activate the short operation	
Datapoint type short operation object	**1 bit** 1 byte scaling 1 byte unsigned 2 bytes float 4 bytes unsigned 4 bytes float
Here the Datapoint type for the short operation object can be selected.	
Event on short operation	**Toggle** On Off
A telegram with one of the above options (if DPT=1 bit where Toggle = opposite to the objects value) as its useful data will be sent when opening the contact before the time for long operation has elapsed. By changing the DPT the value to be sent can be introduced in an input field and the possible range depends on the DPT selection. For 2 byte float values the introduced value will be multiplied by 0.1 in order to send decimal values.	
LONG OPERATION	No **Yes**
This parameter is to activate the long operation	
Datapoint type long operation object	**1 bit** 1 byte scaling 1 byte unsigned 2 bytes float 4 bytes unsigned 4 bytes float
Here the Datapoint type for the long operation object can be selected.	
Event on long operation	Toggle **On** Off
A telegram with one of the above options as its useful data will be sent when opening the contact after the time for long operation has elapsed.	
Time for long operation	**100 ms** 1 s
This time is to distinguish between short and long operation. When releasing before this time, the short operation event will be executed, and afterwards the event for the long operation will be sent.	
OPENING CONTACT	No **Yes**
(Only for “Switching / value / Short + Long operation advanced”) This parameter is to activate the event for opening the contact after the time for long operation has elapsed.	
Event on opening the contact after long operation	**Toggle** On Off
A telegram with one of the above options (if DPT=1 bit where Toggle = opposite to the objects value) as its useful data will be sent when opening the contact after the time for long operation has elapsed. By changing the DPT the value to be sent can be introduced in an input field and the possible range depends on the DPT selection. For 2 byte float values the introduced value will be multiplied by 0.1 in order to send decimal values. ***Attention! This event will be delayed by 50ms and sent using the same object as for long operation***	
Delay of telegram	**No** At short operation At long operation

	At opening contact At all operations
The telegram can be delayed from 1 to 255s for any of the above options.	
Cyclic sending	**No** Short operation Opening contact after long operation Last operation
The telegram can be repeated cyclically for any of the above options. Whether or not the cyclic sending can be stopped with by enabling and/or disabling the input can also be configured.	

4.A.2 Parameter page: InX Binary input / Dimming

Parameter	Settings
Type of input	Dimming
Select this option to dim a light connected to a KNX dimming actuator	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms
This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Attention! For 1 byte absolute dimming use the Sequence function	
Monitor input open circuit / Doubling inputs	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X No alarm = Toggle, Alarm = X
By selecting this function the inputs can be supervised in order to generate an alarm if the input connexion has been cut (only open circuit will generate an alarm). To do this a 2,7k Ohm resistor must be connected to the end of the input line. With the above options one can select what value (nothing, Off, On, Toggle) should be sent with an open circuit alarm and also what value (nothing, Off, On, Toggle) when the alarm goes away.	

4.A.2.1 Parameter page: Dimming

Parameter	Settings
Function of input	Off / darker On / brighter **Toggle brighter / darker**
Select here the function of the input from one of the above options	

4.A.2.2 Parameter page: Dimming / Toggle brighter / darker

Parameter	Settings
Function of input	**Toggle brighter / darker**
With this selection the opposite event to the last executed/received event will be sent. e.g. Previous event: ON -> next event: OFF Previous event: Dim brighter -> next event: Dim darker And vice versa.	
Dimming direction after switching ON	**Darker** Brighter
After sending a ON with the 1 bit object, the next dimming event (4 bit dimming object) will send the parametrized dimming step with dimming direction equal to “Darker”	
Time for long operation	**100 ms** 1 s
This time is to distinguish between short and long operation. When releasing before this time, the 1 bit ON/OFF short operation event will be executed. When reaching this time the 4 bit dimming long operation event will be sent and afterwards when releasing either a stop telegram or not will be sent depending on the next parameter.	

Dimming step	**1 step (100%)** 2 steps (50%) 4 steps (25%) 8 steps (12,5%) 16 steps (6,25%) 32 steps (3,12%) 64 steps (1,6%)
A dimming command, relative to the current brightness setting, is transmitted to the dimming actuator using the relative dimming object DPT_Control_Dimming. Bit 3 of the useful data determines whether the addressed device dims down or up compared to the current brightness value. Bits 0 to 2 determine the dimming step. The smallest possible dimming step is 1/64th of 100 % (1 % in the ETS group monitor).	
Send stop telegram when opening contact	No **Yes**
By selecting this option a stop telegram will be sent when releasing after passing the “time for long operation”	
Cyclic sending	**No** Yes
The telegram will be repeated cyclically (with a configurable frequency), but only during the time the contact is closed.	

4.A.2.3 Parameter page: Dimming / Off / darker
4.A.2.4 Parameter page: Dimming / On / brighter

Parameter	Settings
Function of input	Off/ darker On / brighter
Select the function of the input to switch ON with a short operation and dim brighter with a long operation or switch OFF with a short operation and dim darker with a long operation	
Time for long operation	**100 ms** 1 s
This time is to distinguish between short and long operation. When releasing before this time, the 1 bit ON/OFF short operation event will be executed, and afterwards the 4 bit dimming long operation event will be sent.	

Dimming step	**1 step (100%)** 2 steps (50%) 4 steps (25%) 8 steps (12,5%) 16 steps (6,25%) 32 steps (3,12%) 64 steps (1,6%)
A dimming command, relative to the current brightness setting, is transmitted to the dimming actuator using the relative dimming object DPT_Control_Dimming. Bit 3 of the useful data determines whether the addressed device dims down or up compared to the current brightness value. Bits 0 to 2 determine the dimming step. The smallest possible dimming step is 1/64th of 100 % (1 % in the ETS group monitor).	
Send stop telegram when opening contact	No **Yes**
By selecting this option a stop telegram will be sent when releasing after passing the “time for long operation”	
Cyclic sending	**No** Yes
The telegram will be repeated cyclically (with a configurable frequency), but only during the time the contact is closed.	

4.A.3 Parameter page: InX Binary input / Shutter

Parameter	Settings
Type of input	Shutter
Select this option to control a shutter connected to a KNX shutter actuator	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	

Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms
This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Monitor input open circuit / Doubling input	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X No alarm = Toggle, Alarm = X
By selecting this function the inputs can be supervised in order to generate an alarm if the input connexion has been cut (only open circuit will generate an alarm). To do this a 2,7k Ohm resistor must be connected to the end of the input line.	

4.A.3.2 Parameter page: Shutter / Blind

Parameter	Settings
Event on short operation	**Stop / step up** Stop / step down Toggle stop / step Up Down Toggle up / down
Here the event for the short operation can be assigned. Take note that any of the events can be configured, unlike most KNX shutter/blind sensors.	
Event on long operation	Stop / step up Stop / step down Toggle stop / step **Up** Down Toggle up / down
Here the event for the long operation can be assigned. Take note that any of the events can be configured, unlike most KNX shutter/blind sensors.	
Time for long operation	**100 ms** 1 s
This time is to distinguish between short and long operation. When releasing before this time, the short operation event will be executed, and afterwards the event for the long operation will be sent. Take note that any of the events can be configured for both short and long operation and therefore the objects only indicate the event and not if it is for short or long. I.e. If event for short operation = UP and event for long operation = Down, the "[InX] Blind stop/step" object will never send a telegram.	
Slat time push button	**No** Yes
This is to send a stop telegram after long operation and when releasing within the parametrized time. After this time no telegram will be sent This time should be longer than the total slat time configured in the shutter/blind output channels.	
Waiting time to change slat direction (between short step actions) * Only for Toggle	**100 ms** 1 s
This time is essential to move the slats (with repeated short events) in the same direction when "Toggle ..." is selected. With short step actions longer than this time the next short event will be the inverted action.	
Attention! This time must be longer than the time configured for long operation * Only for "Event on short operation" = Toggle up / down	

4.A.4 Parameter page: InX Binary input / KNX Scene

Parameter	Settings
Type of input	KNX Scene
This type of input selection assigns the input to be a standard KNX 8 bit DPT_Scene_Control sensor.	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with a ON telegram and to disable with an OFF telegram or vice versa.	

Execute scene after bus recovery	**No** Yes
With this option the scene will be executed (the initial sending delay can be adjusted in the general setting tab) on bus voltage recovery.	
Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms
This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Monitor input open circuit / Doubling input	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X No alarm = Toggle, Alarm = X
By selecting this function the inputs can be supervised in order to generate an alarm if the input connexion has been cut (only open circuit will generate an alarm). To do this a 2,7k Ohm resistor must be connected to the end of the input line.	

4.A.4.A.1 Parameter page: KNX Scene

Parameter	Settings
Scene number	**Scene 1** … Scene 64
The scene number to be sent can be configured here. Scene 1 = value 0, Scene 2 = value 1 and so forth up to value Scene 64 = value 63.	
Save scene with long operation	No **Yes**
With this selection the scene can be saved. Saving Scene 1 will send the value 128, Scene 2 sends value 129 and so forth up to Scene 64 sends value 191 to the bus.	
Time for long operation	**100 ms** 1 s
This time is to distinguish between short and long operation. When releasing before this time, the scene will be executed, and afterwards the scene will be saved.	

4.A.1.5 Parameter page: InX Binary input / Multiple operations

Parameter	Settings
Type of input	Multiple operations
With this option more than one telegram can be sent with the same input depending on the number of pulses.	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with a ON telegram and to disable with an OFF telegram or vice versa.	
Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms
This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Monitor input open circuit / Doubling input	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X No alarm = Toggle, Alarm = X
By selecting this function the inputs can be supervised in order to generate an alarm if the input connexion has been cut (only open circuit will generate an alarm). To do this a 2,7k Ohm resistor must be connected to the end of the input line.	

4.A.5.1 Parameter page: Multiple operations / Operation 1…5

Parameter	Settings
Multiple operation 1 (1…5)	No **Yes**
A total of 5 multiple operation can be activated one by one by selecting yes in each one.	

Number of pulses	**1 pulse** … 10 pulses
The number of pulses in the input to execute an event as configured in the next parameters	
Datapoint type of output	**1 bit** 1 byte unsigned 1 byte scaling 2 bytes float
Here the Datapoint type for the "[InX] Multiple op. X pulses"] object can be selected.	
Action on X pulses	On Off **Toggle**
A telegram with one of the above options as its useful data will be sent as the Action on the above configured number pulses.	
Maximum time between pulses	**500 ms** 1 s 2 s 5 s 10 s
For the pulses to be counted, the time between the consecutive pulses may not exceed this parametrized maximum time. Should the time between two consecutive pulses exceed this time, this last pulse and all the following pulses will not be taken into account. It will only start to execute the pulses again once all other multiple operations for this input has been executed.	
Condition for sending value	**Only evaluate last executed pulse operation** Evaluate immediately when operations = pulses
Configure here the sending condition of the output. When "Only evaluate last executed pulse operation" has been selected, the output object will only be sent when the last pulse (when the maximum time between pulses has elapsed) is equal to the number of configured pulses. When "Evaluate immediately when operations = pulses" has been selected, when the number of operations equals the number of pulses, the output will be immediately sent. It will not wait for the last pulse (when the maximum time between pulses has elapsed) to be executed.	

4.A.5.2 Parameter page: Multiple operations / Long operation

Parameter	Settings
Long operation	No **Yes**
This activates the long operation	
Time for long operation	**100 ms** 1 s
This time is to distinguish between pulses and long operation. When releasing before this time, a pulse is counted, and afterwards event for long will be executed.	
Datapoint type for long operation output	**1 bit** 1 byte unsigned 1 byte scaling 2 bytes float
Here the Datapoint type for the "[InX] Multiple op. long object" can be selected.	
Event on long operation	**Toggle** On Off
A telegram with one of the above options as its useful data will be sent when opening the contact after the time for long operation has elapsed.	

4.A.6 Parameter page: InX Binary input / Flashing

Parameter	Settings
Type of input	Flashing
The input can be used to flash ON and OFF with different ON and OFF times.	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with a ON telegram and to disable with an OFF telegram or vice versa.	
Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms

This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Monitor input open circuit / Doubling input	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X No alarm = Toggle, Alarm = X
By selecting this function the inputs can be supervised in order to generate an alarm if the input connexion has been cut (only open circuit will generate an alarm). To do this a 2,7k Ohm resistor must be connected to the end of the input line.	

4.A.6.1 Parameter page: Flashing

Parameter	Settings
Flashing	Close = flash, open = nothing Close = nothing, open = flash **Close = flash, open = stop** Close = stop, open = flash Both = start flashing
Select here with which operation (by opening the contact or closing the contact) the flashing should start and stop. Take into account that the flashing will only start if the contact is opened or closed while the device has bus voltage. Should the contact be closed while there is no bus voltage, and the bus voltage recovers afterwards, then the flashing will neither start nor stop.	
ON duration	1 s 5 s 10 s 1 m 5 m 10 m 1 h
The ON duration can be configured here	
OFF duration	1 s 5 s 10 s 1 m 5 m 10 m 1 h
The OFF duration can be configured here	
Number of repetitions (65535 = always flashing)	65533
This is the number of repetitions the ON/OFF flashing sequence should perform. 0 = No repetitions and 65535 = always flashing.	
Stop flashing	**No** At disabling input At disabling and enabling input
The flashing can be stopped either only at disabling or both for enabling and disabling the input.	

4.A.7 Parameter page: InX Binary input / Sequence

Parameter	Settings
Type of input	Sequence
With this option loads can be sequentially switched ON or OFF. This can be used to have for instance more or less lights ON and thus create the illusion of “dimming” the lights with normal switching actuators.	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms
This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Monitor input open circuit / Doubling input	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X No alarm = Toggle, Alarm = X

By selecting this function the inputs can be supervised in order to generate an alarm if the input connexion has been cut (only open circuit will generate an alarm). To do this a 2,7k Ohm resistor must be connected to the end of the input line.

4.A.7.1 Parameter page: Sequence

The sequence is to switch from one to four output objects sequentially ON or OFF. The sequence is triggered with the rising edge of the input.

Parameter	Settings
Datapoint type of sequence objects	**1 bit** 1 byte unsigned 1 byte scaling 2 bytes float
The datapoint type of the sequence objects can be selected here.	
Number of sequence objects	4
The number of the sequence object can be selected here.	
Type of sequence	Single **Multiple**
The type of the sequence can be selected here. When selecting “Single” only one sequence output object is ON at a time and when selecting “Multiple” more than one object can be ON at a time.	
Multiple (switch sequentially output objects ON)	Incremental ON loop Incremental ON Decremental OFF Decremental OFF loop **Toggle pause** Toggle
Select here in which order the output objects should be switched. Incremental ON loop: 1>1+2>1+2+3>1+2+3+4>All OFF>1>1+2>1+2+3>… Incremental ON loop: 1>1+2>1+2+3>1+2+3+4>stay in 1+2+3+4 Decremental OFF: 4+3+2+1>3+2+1>2+1>1>OFF>stay in OFF Decremental OFF loop: 4+3+2+1>3+2+1>2+1>1>OFF>4+3+2+1>3+2+1>… Toggle pause: (1>1+2>1+2+3>1+2+3+4>Off>1…) pause > 1,5sec. (4+3+2+1>OFF>4>…) The pause time for “Toggle pause” is equal to 1.5 sec. which means that with short pulses less than 1.5 sec. apart it will sequentially switch ON and after waiting more than this time it will sequentially switch OFF. Toggle: Off>1>1+2>1+2+3>1+2+3+4>1+2+3>1+2>1>Off>1+2>...	
Single (only one object ON at a time)	Incremental loop Incremental **Toggle pause** Toggle Decremental Decremental loop
Toggle pause (1>2>3>4>Off>1…), (4>3>2>1>OFF>4>…) ***Attention! Pause time for “Toggle pause” = 1,5 sec.***	
Incremental loop: 1>2>3>4>Off>1>… Incremental: Off>1>2>3>4>stay in 4 Toggle pause: (1>2>3>4>Off>1>…) pause > 1,5sec. (4>3>2>1>Off>4>…) The pause time for “Toggle pause” is equal to 1.5 sec. which means that with short pulses less than 1.5 sec. apart it will sequentially switch ON (only one at a time) and after waiting more than this time it will sequentially switch OFF. Toggle: Off>1>2>3>4>3>2>1>Off>1>… Decremental 4>3>2>1>stay in Off Decremental loop 4>3>2>1>Off>4>…	
Objects to send	**All objects** Only changed objects
It can be selected whether only changed objects or all objects should be sent on each operation.	
Additional input object to trigger sequence (only ON)	**No** Yes
The sequence can also be triggered from the bus to do the same as if the input was pressed. It will only be triggered with ON telegrams.	

Additional input object to inverse sequence (increment / decrement)	**No** Yes
This activates an object to inverse the selected sequence. If the input is used to increment the sequence, with this object the same sequence can be decremented form the bus. It will only be triggered with ON telegrams.	

4.A.8 Parameter page: InX Binary input / Counter

Parameter	Settings
Type of input	Counter
With this parameter the input can be used as a counter.	
Enable / Disable input	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The input can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Send counter values after bus recovery	**No** Yes
The last counter value can be saved on bus voltage failure and will be sent to the bus (the initial sending delay can be adjusted in the general setting tab) on bus voltage recovery if yes is selected.	
Debounce time	10 ms 20 ms **50 ms** 100 ms 150 ms 200 ms
This parameter is used to set the time the input will be blocked after receiving an input signal. This ensures that the input does not generate unwanted duplicate telegrams.	
Monitor input open circuit / Doubling input	**No** Alarm = 1, No alarm = 0 Alarm = 0, No alarm = 1 Alarm = Toggle, No alarm = X No alarm = Toggle, Alarm = X
By selecting this function the inputs can be supervised in order to generate an alarm if the input connexion has been cut (only open circuit will generate an alarm). To do this a 2,7k Ohm resistor must be connected to the end of the input line.	

4.A.8.1 Parameter page: Counter / No / Upward / Backward

Parameter	Settings
Counter	**No** Upward Backward
There two types of counters; Upward = counts up on each trigger event and Backward = counts backward on each trigger event	

Parameter	Settings
Counter	Upward
Counts up on each trigger event	
Data point type of counter	1 byte unsigned 2 bytes unsigned **4 bytes unsigned**
Here the datapoint type for the counter can be selected. Usually, a Switching counter has a 4 bytes unsigned (default option) value. But 1 and 2 bytes unsigned can also be configured for the purpose of showing the value in info displays, which cannot display 4 bytes unsigned values. Attention: Should the counter be programmed with one DPT and in a later stage the DPT is changed the conter value will be overwritten to zero or to the “Initial value counter”	
Count number of triggers on	**Rising edge** Falling edge Rising and falling edge
Decide here the trigger events to increase or decrease the counter. With rising edge the counter will only be triggered when closing the input. With falling edge the counter will only be triggered when opening the input. And With rising and falling edge the counter will be triggered both when closing and opening the input.	
Additional inputs object to trigger counter	**No** Only with ON Only with OFF

	Both
The counter can also be triggered from the bus. Depending on this parameter the counter will be triggered with ON telegrams, OFF telegrams, or with both.	
Initial value counter	**No** Yes
Here the initial different starting value of the counter can be configured. After downloading with the ETS this value will only be overwritten if the new starting value is changed. Take into account that the additional counter will also be reset. Practical example: should the actuator be installed in an existing installation, where the load connected to the current channel has already a known number of switching operations, this information can be used as the "New starting value". But in a later stage, if some other parameter in the actuator must be changed and downloaded, the new current counter value will not be overwritten.	
Threshold value ***Attention! 0 = Deactivated***	**0**
Here you can enter the number of switching operations that will trigger the 1 bit alarm object of the current channel. So, this alarm object will be activated and send a "1" to the bus as soon as the switching counter passes this threshold. Attention: this alarm will also be sent to the bus immediately after bus recovery. Should the conversion factor be activated and set to be for example "Several triggers increases 1 step" = 3, and the threshold value is set to 5 then the sequence will be as follows: 0,0,1,1,1,2,2,2,3,3,3,4,4,4,5,… The alarm is sent in the first 5 after 15 pulses.	
Object for reading / writing the threshold value	**No** Only readable Readable and writeable
With this option the threshold value can be read and/or changed from the bus. Only readable: this option will activate an unsigned counter object, which can be read by the ETS/other KNX devices. Readable and writable: this option will activate an unsigned counter object, which can be read and overwritten by the ETS/other KNX devices. This is meant to allow changing the threshold value with, for instance, a visualization.	
Reaction on overflow (Max. value of DPT) ***Attention! Both counter & alarm objects will be set to zero***	**Reset to 0 and start again** Stay at maximum
Important note: the overflow must not be mistaken with the threshold value, since they are two totally different concepts: - An overflow is reached when the object value exceeds the maximum value of the selected data point type. For example, the maximum value of a 1 byte unsigned value is 255; therefore, the overflow is reached when the object value exceeds 255. - On the other hand, the threshold refers to any given value of your choice that is valid for this DPT. Reset to 0 and start again (default option): when then overflow is reached, the object will start counting from 0 again. Attention! In this case the alarm object will also be set to zero, otherwise one would not know if the threshold has newly been reached or not. Stay at maximum: in the event of the overflow being reached, the object will stop at the maximum value of the DPT.	
Additional functions	**No** Yes
In order to keep the application program as easy as possible, only the main and most important functions are displayed at first sight. You will often find the possibility to activate the Additional or Advanced Functions, which disclose new functions that are not essential, but can be very useful.	

Parameter	Settings
Counter	Backward
Counts backward on each trigger event	
Data point type of counter	1 byte unsigned 2 bytes unsigned **4 bytes unsigned**
Here the datapoint type for the counter can be selected.	

Usually, a Run hour counter has a 4 bytes unsigned (default option) value. But 1 and 2 bytes unsigned can also be configured for the purpose of showing the value in info displays, which cannot display 4 bytes unsigned values	
Count number of triggers on	**Rising edge** Falling edge Rising and falling edge
Here can be decided when the counter should be triggered. When closing the contact (Rising edge), opening the contact (Falling edge) or both (Rising and falling edge)	
Additional inputs object to trigger counter	**No** Only with ON Only with OFF Both
The counter can also be triggered from the bus with the above options.	
Initial value counter ***Attention! After programming this value will only be overwritten if the new starting value is changed***	**800**
Here the initial different starting value of the counter can be configured from which the counter will count back. It will send a 1 bit alarm telegram with the value "1" when reaching the value zero. Attention! This value will never be sent. The 1st value sent will be the first decreased value. After downloading with the ETS this value will only be overwritten if the new starting value is changed. Take into account that the additional counter will also be overwritten if the main counter is overwritten. Should the conversion factor be activated and set to be for example "Several triggers decreases 1 step" = 3, and the "Initial value switching counter" is set to 5 then the sequence will be as follows: 444,333,222,111,000, and only at the last 0 the alarm will be sent.	
Reaction on reaching zero	Stay at zero **Reset to initial value and start again**
Stay at zero: once the counter reaches 0, it will stay there until it has been reset.	
Reset to initial value and start again (default option): once the counter reaches 0, it will start counting back again starting from the initial value of the switching counter (as parameterized in the previous option).	
Additional functions	**No** Yes
In order to keep the application program as easy as possible, only the main and most important functions are displayed at first sight. You will often find the possibility to activate the Additional or Advanced Functions, which disclose new functions that are not essential, but can be very useful.	

4.A.9 Parameter page: Counter / Additional functions

Parameter	Settings
Cyclic sending of counter value	**No** Yes
With this option the counter values can be sent cyclicly which can have a frequency from 10 sec. up to 255 hours.	
Counter values are sent to the bus every: (Triggers)	**1**
Enter here the number of switching operations that be executed before the counter sends its value to the bus. This option is meant to reduce the bus traffic. For instance, if you enter a "50", the counter will send its first value whenever the accumulated switching operations of the channel amount to 50 and will then send the value 50 to the bus (50, 100, 150, 200, 250…).	
Conversion factor	**None** Several triggers increases 1 step 1 trigger increases several steps
None (default option): for each switching operation of the channel, the counter increases 1 step. Several triggers increases 1 step: define here the number of triggers that must be received for the counter to increase 1 step. Should it be set to the value 10, then only when triggers received amount to 10, will the counter increase 1 step. 1 trigger increases several steps: define here the step increment for each trigger received. For example, if it is set to 50, after 50 triggers received, the counter will have increased 50 x 10 (= 500) steps.	

Send last value of counter at reset by counter object	**No** Yes

No (default option): if you reset the counter by using the 1 bit reset object, the last value of the counter will not be sent to the bus by the counter object. Instead, a "0" will be sent to indicate it has been reset.

Yes: if you reset the counter by using the 1 bit reset object, the counter object will send its current value before reset to the bus and afterwards it will not reset to 0 but stay at its last value. Only at the next counter step, will the first counter step be sent to the bus. Thus the counter will never have the value "0".

Additional object to store last value of counter on reset	**No** Yes Yes and send

No (default option): no additional object to store the last value of the counter on reset will be activated.

Yes: an additional object to store the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse).

Yes and send: an additional object to store and send the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). This value will then be sent after reset using this additional object.

Activate additional counter * Only with counter Upward	**No** Yes

The additional counter counts the same input signal.

It can be used to inform about, for example, the daily value. To do this a time switch is needed to reset this additional counter once a day (or any other desired interval)

Additional upwards counter	Rising edge **Falling edge** Rising and falling edge

Here can be decided when the additional counter should be triggered. When closing the contact (Rising edge), opening the contact (Falling edge) or both (Rising and falling edge)

Additional upwards counter initial value	**0**

Here the initial different starting value of the counter can be configured from which the counter will count.

After downloading with the ETS this value will only be overwritten if the new starting value is changed.

Reaction on overflow (Max. value of DPT)	**Reset to 0 and start again** Stay at maximum

Important note: the overflow must not be mistaken with the threshold value, since they are two totally different concepts:

- An overflow is reached when the object value exceeds the maximum value of the selected data point type. For example, the maximum value of a 1 byte unsigned value is 255; therefore, the overflow is reached when the object value exceeds 255.
- On the other hand, the threshold refers to any given value of your choice that is valid for this DPT.

Reset to 0 and start again: when then overflow is reached, the object will start counting from 0 again. Attention! In this case the alarm object will also be set to zero, otherwise one would not know if the threshold has newly been reached or not.

Stay at maximum: in the event of the overflow being reached, the object will stop at the maximum value of the DPT.

Additional object to store last value of counter on reset	**No** Yes Yes and send

No: no additional object to store the last value of the counter on reset will be activated.
Yes: an additional object to store the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse).
Yes and send: an additional object to store and send the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). This value will then be sent after reset using this additional object.

4.B. Parameter page: ANALOG INPUTS

There are 6 inputs which can be configured to receive binary (push buttons, window contacts, water leakage sensor…) and analog signals (movement detector, temperature sensor and monitored input…)

Parameter	Settings
Input 1...6	No function **Binary input** Movement detector Temperature sensor

4.B.1 Parameter page: InX Movement detector

The input of the actuator can be used to connect any conventional binary movement detector with a N.O. relay output or an analogue detector from Ipas (not yet available) and convert it into a fully functional KNX movement detector. It has up to two channels: one lighting channel and a HVAC channel.

Parameter	Settings
Type of movement detector	**Analog & Bin. detector. Time in parameter** Only binary detector N.O. Time in detector
Attention! For binary detector, manually adjust the pulse time in external detector as short as possible!	
The type of detector basically determines whether or not the time should be adjusted in the detector or in the application program. When selecting "Only binary detector N.O. Time in detector", there is no detection time parameter in the ETS application program and the time must be set in the detector (usually with a small time adjustment screw). When selecting "Analog & Bin detector. Time in parameter", the time can be adjusted in the application program. For the binary detector the pulse time should be adjusted manually with the small time adjustment screw on the detector to be as short as possible since the time starts counting the moment the relay opens.	

4.B.1.1 Parameter page: InX Movement detector / Analog & Bin. detector. Time in parameter

When selecting "Analog & Bin detector. Time in parameter", the time can be adjusted in the application program. For the binary detector the pulse time should be adjusted manually with the small time adjustment screw on the detector to be as short as possible.

With this selection both the lighting and HVAC channels will be available. (With "Only binary…" only the lighting channel can be used.)

Both the lighting channel and the HVAC channel can be activated.

Parameter	Settings
Lighting channel	No **Yes**
This parameter is used to activate the lighting channel tab and all its parameters.	
HVAC channel	**No** Yes
This parameter is used to activate the HVAC channel tab and all its parameters.	
Blocking time after end of detection	500 ms
- Factor (1…255)	4
The detector can be blocked for a configurable time after end of detection; this time can be set here. This could be important depending on the load to be switched by the detector. Passive IR movement detectors detect moving heat, the detector detects any heat source which crosses the IR sectors of the detector. Since a light bulb is hot when switched on and cools down when switched off, it also generates moving heat and thus the detector can falsely interpret this to be a movement, after which the light would switch on again. This time is meant to avoid this conflict and should be adjusted depending on the heat generated by the bulb to be controlled and the distance to the detector.	

4.B.1.1.1 Parameter page: InX Movement detector / Analog & Bin. detector. Time in parameter / Lighting tab

Parameter	Settings

µBrick Actuator Series

Datapoint type lighting channel output	**1 bit** 1 byte scaling 1 byte unsigned 2 bytes float 4 bytes unsigned 4 bytes float
The DPT of the output object for the lighting channel can be set to any of the above DPTs.	
Event at beginning of detection	Nothing **Value**
- Value to send	1
Here the value to be sent to the bus at the beginning of detection can be set. The option to send nothing is also available.	
Event at end of detection	Nothing **Value**
- Value to send	0
Here the value to be sent to the bus at the end of detection can be set. The option to send nothing is also available.	
Total time after last detection (Time starts when relay opens)	**1 s** 10 s 1 min 10 min 1 h
- Factor (1…255)	60
This is the time which must elapse without having received a detection pulse in the input from the connected detector, for it to trigger the event on end of detection.	
Cyclic sending	**No** Only on detection Only at the end of detection Both
Here one can choose the cyclic sending of the output telegram to be only on detection, only at end of detection or in both cases.	
Brightness dependent switching	**No** Analog detector – light sensor External object
The detector can switch the light dependent on the brightness value. This value can be received from the analogue measurement from the light sensor of the detector to determine the LUX level, or from a KNX light sensor by sending its value to the external object of the input.	

Threshold (detection is enabled when brightness is lower than)	**80**
This option is only available when “Analog detector – light sensor” or “External object” have been selected. When selecting “Analog detector – light sensor” the input will read the analogue value from the light sensor of the detector to determine the LUX level and it will block the detector if the brightness is higher than the parametrized threshold value set here. When selecting “External object” the value can be sent from a KNX light sensor to the external object of the input. It can then block the detector if the brightness is higher than the parametrized threshold value set here. For example during the day (high LUX level) the detector is blocked, as it gets dark enough to detect, (i.e. lower than the parameter value) it should enable the detector and stay enabled until the light level increases above the threshold value again.	
Enable / disable lightning channel	**No** Yes
It is possible to block the lighting channel with one or even two “Enable / disable …” objects. These objects are purely trigger objects to enable or disable the detector and it is NOT necessary to enable or disable both objects in order to enable or disable the detector. The last action received on these objects will determine the state of the detector. Therefore, they will not inform about whether or not the detector is blocked. For this purpose there is an additional status object to inform about whether the detector is enabled or not. *Practical example*: a very typical requirement in a KNX installation is to be able to block the light in an ON state (for instance, during a meeting) but it is as important to block the light in an OFF state. (For instance, projector mode). That is why there are two objects to block the detector, each with a different behaviour when blocking and unblocking.	
- Reaction on bus voltage recovery	**Enable** Disable Last object status
Here we can configure whether the lighting channel of the detector should be enabled or not on bus voltage recovery. It can also return to the status before bus failure.	

Enable lighting channel by object 1	**En = 1 / Dis = 0** En = 0 / Dis = 1
Attention! The "MD lighting Disable 1&2" objects don't indicate the "disabled" status. The last object updated sets the state (independent of the other object)	
Here you can configure the value to enable or disable the detector with the first enable object.	
Send telegram when enabling lighting channel	**Don't send** Value
- Value to send	**1**
Use this parameter to set the value to be sent to the bus when enabling the channel with the first enable object. This telegram will be sent on each enable telegram (no need to change from the disabled state)	
Send telegram when disabling lighting channel	**Don't send** Value
- Value to send	**0**
Set here the value to be sent to the bus when disabling the channel with the first enable object. This telegram will be sent on each disable telegram (no need to change from the enabled state)	
Enable lighting channel by object 2	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
Attention! The "MD lighting Disable 1&2" objects don't indicate the "disabled" status. The last object updated sets the state (independent of the other object)	
Configure with this parameter the value to enable or disable the detector with the second enable object.	
Send telegram when enabling lighting channel	**Don't send** Value
- Value to send	**1**
Use this parameter to set the value to be sent to the bus when enabling the channel with the second enable object. This telegram will be sent on each enable telegram (no need to change from the disabled state)	
Send telegram when disabling lighting channel	**Don't send** Value
- Value to send	**0**
Set here the value to be sent to the bus when disabling the channel with the second enable object. This telegram will be sent on each disable telegram (no need to change from the enabled state)	

4.B.1.1.2 Parameter page: InX Movement detector / Analog & Bin. detector. Time in parameter / HVAC tab

Parameter	Settings
Datapoint type HVAC channel output	**1 bit** 1 byte scaling 1 byte unsigned 2 bytes float 4 bytes unsigned 4 bytes float
The DPT of the HVAC output object can be selected here.	
Initial waiting time for HVAC activation (time starts when relay closes)	1 s 10 s **1 min** 10 min 1 h
- Factor	3
This is the initial waiting time which must elapse for the HVAC channel of the detector to detect movement. This time starts to count when the relay of the external detector closes. Should a person only go into the detection range of the detector and immediately thereafter go out again, the HVAC channel of the detector will not detect movement. Thus the HVAC system will only be switched to the desired operating mode if someone goes into the room and stays in this room longer than the configured time. Due to the fact that this is usually a long time (3 minutes default parameter) and passive IR detectors are not perfect (they don't detect always all small movements, they only detect moving heat objects), a special algorithm has been implemented to determine if someone is staying in the room or not. Explanation of this algorithm by means of an example: Let's say the "Initial waiting time…" is set to be 10 min. Then during the first 50% (5 min.) of the time, the detection pulses are ignored. Thereafter, during the rest of the time the input	

should detect detection pulses within a time window equal to 30% of the full "Initial waiting time…" (30% of 10 min. = 3 min.; so, every 3 minutes), otherwise the time will reset to the initial 10 minutes and the process will start all over again. In other words, in this example: - During the first 5 minutes it will not detect any pulses. - From minute 5 to minute 8: the input must detect at least one pulse. If the pulse is received, it will reset the 30% timer. - If the input detects a pulse at minute 6, then the input must detect the next pulse from minute 6 to minute 9. - If the input detects a pulse at minute 7, then the input must detect the next pulse from minute 7 to minute 10. - Then, if the input detects a pulse just after minute 7, then the HVAC channel will be activated on minute 10 even if no pulse is afterwards received.	
Event at beginning of detection	Nothing **Value**
- Value to send	1
Configure here the value to be sent to the bus at the beginning of detection of the HVAC channel. The option to send nothing is also available.	
Event at end of detection	Nothing **Value**
- Value to send	0
Configure here the value to be sent to the bus at the end of detection of the HVAC channel. The option to send nothing is also available.	
Total time after last detection (Time starts when relay opens)	1 s 10 s **1 min** 10 min 1 h
- Factor (1…255)	30
This is the time which must elapse without any detection for the input to send the event at end of detection. This time starts to count at the beginning of detection and thus when the initial waiting time ends.	
Cyclic sending	**No** Only on detection Only at the end of detection Both
Here one can choose the cyclic sending of the output telegram to be only on detection, only at end of detection or in both cases.	
Enable / disable HVAC channel by object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The HVAC channel can be enabled or disabled with a 1 bit object. Here can be decided to enable with a 1 and disable with a 0 or vice versa.	
- Reaction on bus voltage recovery	**Enable** Disable Last object status
Whether the HVAC channel of the detector will be active or not on bus voltage recovery can be configured here. On bus voltage recovery the HVAC channel can be enabled, disabled, or have the same state as before the bus failure depending on the above selection. **Enable:** the HVAC channel will be enabled. **Disable:** the HVAC channel will be disabled. **Last object status:** the status of the Enable object will be saved in the actuator's non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will set the object as it was before the bus failure.	
Send telegram when enabling HVAC channel	**Don't send** Value
- Value to send	**0**
Use this parameter to define the value to be sent to the bus when enabling the HVAC channel with the HVAC enable object.	
Send telegram when disabling lighting channel	**Don't send** Value
- Value to send	**0**
Use this parameter to define the value to be sent to the bus when disabling the HVAC channel with the HVAC enable object.	

4.B.1.2 Parameter page: InX Movement detector / Only binary detector N.O. Time in detector

When selecting "Only binary detector N.O. Time in detector" there is no detection time parameter in the ETS ap-

plication program and the time must be set in the detector (usually with a small time adjustment screw). For this reason, only the lighting channel can be used.

All the parameters of the lighting channel are the same as in the previous type of movement detector, but without the parameter to adjust the time after last detection.
There is no HVAC channel.

4.B.2 Parameter page: InX Temperature sensor

When selecting Temperature sensor the Ipas NTC Temperature Sensor should be connected between the "C" common terminal and the input.

Parameter	Settings
Attention! If no temperature sensor is connected to the input, the first source will be ignored	
First source temperature value	**2 bytes float** 4 bytes float
The temperature value can be sent either with a 2 bytes float value (most common) or with a 4 byte float value.	
Sensor calibration value (ºC x0,1)	**0**
Here the calibration value can be set in order to higher or lower the measured value which will be sent to the bus.	
Second source temperature value	**No** External object
It is possible to activate an input object to receive the value of a second sensor.	
Datapoint type for external input object	**2 bytes float** 4 bytes float
The external input object for the second sensor can be a 2 or 4 byte float value.	
Datapoint type for weighted output object	**2 bytes float** 4 bytes float
The external weighted output object can be a 2 or 4 byte float value. The value of this object is a weighted value between the two sensor sources of the input.	
Weighted source % (first – second)	10 - 90 20 - 80 30 - 70 40 - 60 **50 - 50** 60 - 40 70 - 30 80 - 20 90 - 10
Establish here the percentage of the first and second source in order to calculate the weighted output value.	
Attention! Only weighted output will be sent	
When 2 sources are used to calculate a weighted value it will send only this weighted output and not the value of the source itself.	
Sending condition	Only readable **On change**
The sending condition can be set to be only on value change or if it should be only readable and thus only answer to read requests.	
Send with changes higher than (ºC x0,1)	**5**
When selecting "On change" the temperature value change (in decimals of a degree centigrade) can be set here in order to generate a new telegram to the bus.	
Cyclic sending	**No** Yes
It is also possible to send the telegram cyclic to the bus. The cyclic rate can be set individually in the next parameter.	
- Base	10 s **1 min** 5 min 10 min 1 h
- Factor (1…255)	**1**
Temperature input supervision	**No** First source Second source Both
See next section.	
Send all status telegrams after bus recovery	**No** Yes
Attention! Activate "Behaviour at bus recovery" & set delay in "General Settings"	
All temperature status values can be sent to the bus after the initial delay (if activated) after bus recovery.	

4.B.2.1 Parameter page: InX Temperature sensor / Temperature input supervision

It is possible to supervise only the first source, only the second source or both sources.

4.B.2.1.1 Parameter page: InX Temperature sensor / Temperature input supervision / First source

Parameter	Settings
Attention! First source failure will immediately send an error telegram	

When selecting first source it will supervise the input terminal to have a valid value. Should an invalid value be detected (e.g. should the input be disconnected i.e. input wire breaks, short circuits etc.) it will generate a 1 bit alarm message.	
Behaviour with source failure	**Only use other sensor (without weight)** Use last value
Select here the behaviour with source failure. When selecting “Only use other sensor (without weight)”, the invalid source will be ignored and only the valid source value will be sent as is (without applying any weighting formula) with the weighted output object.	

4.B.2.1.2 Parameter page: InX Temperature sensor / Temperature input supervision / Second source

Parameter	Settings
Second source cyclic supervision time	10 s **1 min** 5 min 10 min 1 h
- Factor	1
When selecting second source it will supervise if the second source input object receives a telegram within the cyclic supervision time. If no telegram has been received within this time a 1 bit alarm message will be sent to the bus.	
Behaviour with source failure	**Only use other sensor (without weight)** Use last value
Select here the behaviour with source failure. When selecting “Only use other sensor (without weight)”, the invalid source will be ignored and only the valid source value will be sent as is (without applying any weighting formula) with the weighted output object.	

4.B.2.1.3 Parameter page: InX Temperature sensor / Temperature input supervision / Both (sources)

Parameter	Settings
Second source cyclic supervision time	10 s **1 min** 5 min 10 min 1 h
- Factor	1
Attention! First source failure will immediately send an error telegram	
When selecting both, it will supervise both the input terminal to have a valid value, and if the second source input object receives a telegram within the cyclic supervision time. Should an invalid value be detected in the first source (e.g. should the input be disconnected i.e. input wire breaks, short circuits etc.) or if no telegram has been received in the second source input object within the cyclic supervision time a 1 bit alarm message will be sent to the bus.	
Behaviour with source failure	**Only use other sensor (without weight)** Use last value
Select here the behaviour with source failure. When selecting “Only use other sensor (without weight)”, the invalid source will be ignored and only the valid source value will be sent as is (without applying any weighting formula) with the weighted output object.	

µBrick Actuator Series

5. Parameter page: GENERAL SETTINGS / OUTPUTS

Parameter	Settings
Outputs	No **Yes**
The outputs of the actuator are by default activated. Nevertheless, this device can also be used as an advanced controller module for logic functions, timers, etc. In this case, you can deactivate the outputs totally and completely hide all their options and objects by selecting “No”.	

Parameter page: OUTPUTS

Parameter	Settings
CHANNEL A … CHANNEL C	**Binnary** Shutter / Blind No function
Each cannel can be configured either as Two Binary Channels or One Shutter/Blind Channel. If the channel is not meant to be used, you can hide all its options and tabs by choosing the “No Function” option.	
Central ON/OFF, UP/DOWN object	**No** One common object Two separate objects
In order to do a classic KNX “Central function”, this actuator has a specific option that allows for all the channel actions to be performed at once with only one or two objects. This considerably reduces the amount of group address associations (both meant to ease programmers work load, but also to reduce the actuator’s association table). Before we configure the function within the channel, we must activate one of the objects. The actuator has 1 or 2 Central ON/OFF, UP/DOWN objects for binary outputs and/or shutter: o 1 common object = “Central switching/move blind” o 2 separate objects = “Central switching” + “Central move”	
Manual control	Param Mode + Test Mode Param Mode Test Mode Disable

The µBrick actuator has 2 push buttons and status LEDs on the front side. These buttons can be used to control the current channel according to your selection in this parameter option. Please, see **Annex 1** to learn more about manual control.

In this Parameter menu the behaviour of those push buttons and LEDS can be configured according to the following options:

- **Param Mode + Test Mode (default option)**: both modes will be available.

  When the actuator starts up, it finds itself in Parameter Mode. In order to change to Test Mode, you must press both buttons simultaneously until the LED of the selected channel starts blinking (short blinking action once every second). To go back to Parameter Mode, you have to press both buttons at the same time again until the blinking stops.
- **Param Mode:** only this mode will be available.
- **Test Mode:** only this mode will be available.
- **Disable:** you can also deactivate the Manual Control functionality.

Value for disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The Manual Control functionality can also disabled via an external object. The command used for enabling/disabling this function can be parameterized here.	

5.1 Parameter page: OUTPUTS / Channel A1…X1 (Binary)

Parameter	Settings
Type of contact	**NO-Normally open: ON=close, OFF=open** NC-Normally close: ON=open, OFF=close
Use this parameter option to set whether the output relay closes with ON (“1”) and opens with OFF (“0”) or if it closes with OFF (“0”) and opens with ON (“1”).	
Reaction on bus voltage failure	**Unchanged** ON OFF
Here you can select one of the following reactions**:** if “Unchanged”, whenever the bus voltage fails, the contact stays the same. If you choose ON/OFF, ***as soon as the bus voltage fails, the contact switches on/off (which means, independent of the type of contact, it closes/opens)***	
Reaction on bus voltage recovery	**Unchanged** ON OFF Recovery status before bus failure Timer 1 reaction at ON Timer 2 reaction at OFF

Here you can select one of the following reactions**:**
If “Unchanged”, whenever the bus voltage returns, the contact stays the same.
With ON/OFF, ***as soon as the bus voltage returns, the contact switches on/off (which means, independent of the type of contact, it closes/opens).***
With “Recovery status before bus failure**”,** the status of the output will be saved in the actuator’s non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will switch the output as it was before the bus failure.

Each output has two timer functions. Only the first timer can be assigned to the reaction on bus voltage recovery.

- Timer 1 reaction at ON: the function that has been chosen under “OUTPUTS/Timer 1/REACTION AT ON” will be executed.
- Timer 1 reaction at OFF: the function that has been chosen under “OUTPUTS/Timer 1/REACTION AT OFF” will be executed.

Status	No **Yes**
While the option Yes activates the “Status tab”, No deactivates the “Status tab” and also the “Status object”.	
Advanced functions	No **Yes**
The µBrick Actuator range is also a powerful controller module (logic, timer, counter, etc. module). You can find Advanced Functions: 1) In the General Settings parameter page: this a totally independent controller module, with its own input and output objects, which can work autonomously (no need to be linked to any actuator function). 2) On top of that, the most common advanced functions are also available within each and every channel. The main difference is that these are linked to the channel and cannot be used independent from it. This has the advantage that it is not necessary to use group addresses to link them, making configuration easier.	
Manual control	**No** Yes
The µBrick actuator has 2 push buttons and status LEDs on the front side. These buttons can be used to control the current channel if you select “yes” in this parameter option. Please, see **Annex 1** to learn more about manual control.	

5.1.1 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / Status

Each channel has a separate tab to configure its status parameters, such as the different sending conditions.

Parameter	Settings
Send status telegram	**Only on change** Always Only on change - Inverted Always - Inverted No
Only on change: the status of the output will only be sent whenever the contact switches from on to off or vice versa. **Always:** after reception of each channel-dependent telegram (not only via the "Switching object"), the status will be sent to the bus. **Only on change – Inverted:** the inverted status of the output will only be sent whenever the contact switches from on to off or vice versa. **Always – Inverted:** after reception of each channel-dependent telegram (not only via the "Switching object"), the inverted status will be sent to the bus. **No:** the "Status object" of this channel will be hidden.	
Cyclic sending status telegram	**No** Only ON Only OFF Both ON / OFF
No: the status telegram is only sent once. **Only ON:** if the output changes to ON status, it will send the ON status cyclically. **Only OFF:** if the output changes to OFF status, it will send the OFF status cyclically. **Both ON / OFF:** in both cases (when the output changes to ON or OFF status), it will send the corresponding status cyclically. For these last three options the cyclic sending time can have a base of 10s, 1 min, 5 min, 10 min, 1 hour, and the factor can be from 1 to 255. Should a status telegram be sent (not because of cyclic sending) the cyclic sending time will be reset in order to avoid unwanted duplicate telegrams.	
Delay status telegram	**No** Yes
Depending on the previously configured sending condition, the Status telegram can also be sent to the bus with a time delay.	
Send status telegram at bus recovery	**No** Yes
Attention! Activate "Behaviour at bus recovery" & set delay in "General settings".	
With Yes, the status of the channel will be sent after bus recovery. This initial status telegram can also be sent with a delay, which can be configured in "General Settings/Behaviour at bus recovery" – "Delay for sending all status telegrams" If this delay is set, and the behaviour after bus recovery is set to switch the channel, this switching after bus recovery will not cause a status telegram to be sent to the bus. Only after the initial status delay (as described above) the status telegram will be sent. This delayed sending behaviour is to avoid that all the devices send their status at the same time after bus recovery (even if all outputs are switched at the same time after bus recovery) For example if the delay is set to be 10 seconds and the behaviour after bus return is set to switch the channel ON. Then the channel will be switched ON immediately after bus recovery (this will not cause any status telegrams to the bus) and then 10 seconds later the status telegrams will be sent.	

5.1.2 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS

Parameter	Settings
Central ON/OFF function	**No reaction** Any value = ON Any value = OFF 0 = OFF, 1 = ON 0 = ON, 1 = OFF Any value = Timer 1 reaction at ON 0 = X, 1 = ON 0 = OFF, 1 = X
No reaction: the channel has no reaction when the Central ON/OFF object/s receive/s a telegram. **Any value = ON:** the channel switches ON when the Central ON/OFF object/s receive/s any telegram (no matter whether "0" or "1" is received). **Any value = OFF:** the channel switches OFF when the Central ON/OFF object/s receive/s any telegram (no matter whether "0" or "1" is received). **0 = OFF, 1 = ON:** the channel switches OFF when the Central ON/OFF object/s receive/s a "0" and switches ON when receiving a "1". **0 = ON, 1 = OFF:** the channel switches ON when the Central ON/OFF object/s receive/s a "0" and switches OFF when receiving a "1". **Any value = Timer 1 reaction at ON:** when the Central ON/OFF object/s receive/s any value, the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT ON" will be executed **0 = X, 1 = ON:** the channel has no reaction when the Central ON/OFF object/s receive/s a "0" and switches ON when receiving a "1". **0 = OFF, 1 = X:** the channel switches OFF when the Central ON/OFF object/s receive/s a "0" and has no reaction when receiving a "1".	
Additional object	**No** Inverted Toggle only with 0 Toggle only with 1 Toggle with 0 and 1
No: this option hides the additional object. **Inverted:** if the contact has been configured as normally open (default option), it will switch ON with a "0" and switch OFF with a "1". In other words, it does the opposite to the switching object. **Toggle only with 0:** the output will change its state from OFF to ON or vice versa when receiving "0" (it will ignore the telegram when receiving a "1") **Toggle only with 1:** the output will change its state from OFF to ON or vice versa when receiving "1" (it will ignore the telegram when receiving a "0") **Toggle with 0 and 1:** the output will change its state from OFF to ON or vice versa both when receiving "0" or "1".	
Counters	**No** Yes
There are two counters (one "Run hour" and one "Switching") per channel available, both of which can be configured to count up or down. **No:** this option hides the counter tab and all its objects and options. **Yes:** this option activates the counter tab.	
Scenes	**No** Yes
KNX standard 1 byte scenes: 1 Scene object per output. The advantage of having a Scene object per channel (and not only one for the all the channels) is that with the same Scene number, different scenes can be executed (since they are linked to another push button, with a different group address). Up to 8 scenes can be configured per channel. **No:** this option hides the Scenes tab and all scene related functions and object for the current channel. **Yes:** this option activates the Scene tab, with multiple functions and the Scene object for this channel.	
Timer 1 Timer 2	**No** Yes
There are two timers linked to the current channel and which can run parallel; also, they have their own triggering object each. These timers can be configured to works as ON and/or OFF Delay, Staircase, Delay and staircase, blinking, etc. **No**: the Timer tab and all timer related functions are hidden. **Yes:** the Timer tab and the trigger object will be available, but they have no function assigned and this must be configured in the Timer tab.	
Disable	**No** Yes

Each and every channel have a Disable object, which blocks all other functions of the channel. The behaviour at Disabling/Enabling can be configured per channel. **No:** the Disable object and tab will be hidden. **Yes:** this option activates the Disable object and tab.	
Alarms	**No** Yes
Now, in the Advanced Functions of the current channel, you can configure the behaviour of the channel when the alarm objects receive a telegram. After choosing the “Yes” option, the channel-related Alarms tab will be displayed.	
Manual control	No **Yes**
The µBrick actuator has 2 push buttons and status LEDs on the front side. These buttons can be used to control the current channel if you select “yes” in this parameter option. You can see the exact behaviour of these buttons in OUTPUTS / MANUAL CONTROL	

5.1.2.1 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters

There are two counters (one “Run hour” and one “Switching”) per channel available, both of which can be configured to count up or down.

A) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Run hour counter

Parameter	Settings
Run hour counter	**No** Upward Backward
No: this option hides the Run hour counter tab and all its objects and options. **Upward:** this option is used to count the accumulated time during which the channel has been switched ON. **Backward:** to count down from a configurable initial value.	

A.1) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Run hour counter - UP

Parameter	Settings
Data point type of counter	1 byte unsigned 2 bytes unsigned **4 bytes unsigned**
Usually, a Run hour counter has a 4 bytes unsigned value. But 1 and 2 bytes unsigned can also be configured for the purpose of showing the value in info displays, which cannot display 4 bytes unsigned values.	
Initial value run hour counter	**No** Yes
Attention! After programming this value will only be overwritten if the new starting value is changed.	

This option gives you the possibility to establish an initial value from which the counting will start up. After downloading with the ETS this value will only be overwritten if the new starting value is changed. Practical example: should the actuator be installed in an existing installation, where the load connected to the current channel has already a known number of run-hours, this information can be used as the “New starting value”. But in a later stage, if some other parameter in the actuator must be changed and downloaded, the new current counter value will not be overwritten.	
Run hours threshold value	0
Attention! 0 = Deactivated	
Here you can enter the number of run hours that will trigger the 1 bit alarm object of the current channel. So, this alarm object will be activated and send a “1” to the bus as soon as the Run hour counter passes this threshold. Should the conversion factor be activated and set to be for example “Several run-hours increases 1 step” = 3, and the threshold value is set to 5 then the sequence will be as follows: 0,0,1,1,1,2,2,2,3,3,3,4,4,4,5,… The alarm is sent in the first 5 after 15 pulses. Attention, this alarm will also be sent to the bus immediately after bus recovery.	
Object for reading / writing the threshold value	**No** Only readable Readable and writable
Only readable: this option will activate an unsigned counter object, which can be read by the ETS/other KNX devices. **Readable and writable:** this option will activate an unsigned counter object, which can be read and overwritten by the ETS/other KNX devices. This is meant to allow changing the threshold value with, for instance, a visualization.	
Reaction on overflow (Max. value of DPT)	**Reset to 0 and start again** Stay at maximum
Attention! Both counter & alarm objects will be set to zero	

Important note: the overflow must not be mistaken with the threshold value, since they are two totally different concepts: - An overflow is reached when the object value exceeds the maximum value of the selected data point type. For example, the maximum value of a 1 byte unsigned value is 255; therefore, the overflow is reached when the object value exceeds 255. - On the other hand, the threshold refers to any given value of your choice that is valid for this DPT. **Reset to 0 and start again:** when then overflow is reached, the object will start counting from 0 again. Attention! In this case the alarm object will also be set to zero, otherwise one would not know if the threshold has newly been reached or not. **Stay at maximum:** in the event of the overflow being reached, the object will stop at the maximum value of the DPT.	
Additional functions	**No** Yes
In order to keep the application program as easy as possible, only the main and most important functions are displayed at first sight. You will often find the possibility to activate the Additional or Advanced Functions, which disclose new functions that are not essential, but can be very useful.	

a) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Run hour counter – UP / ADDITIONAL FUNCTONS

Parameter	Settings
Cyclic sending of counter value	**No** Yes
When this function is activated, the corresponding object will not send the telegram once, but repeat it infinitely.	
Counter values are sent to the bus every: (Run hours)	1
Enter here the number of hours that must go by before the counter sends its value to the bus. This option is meant to reduce the bus traffic. For instance, if you enter a “5”, the counter will send its first value whenever the accumulated ON time of the channel has reached 5 hours and will then send the value 5 to the bus (10, 15, 20, 25, 30, 35…).	
Conversion factor	**None** Several hours increases 1 step 1 hour increases several steps
None: for each 1 hour accumulated ON time of the channel, the counter increases 1 step. **Several hours increases 1 step:** define here the number of accumulated ON time (in hours) that must go by for the counter to increase 1 step. **1 hour increases several steps:** define here the step increment for each hour of accumulated ON time. For example, after 8 accumulated ON time hours, the counter will have increased 8 x 10 (= 80) steps.	
Send last value of counter at reset by counter object	**No** Yes
No: if you reset the counter by using the 1 bit reset object, the last value of the counter will not be sent to the bus by the counter object. Instead, a “0” will be sent to indicate it has been reset. **Yes:** if you reset the counter by using the 1 bit reset object, the counter object will send its current value before reset to the bus and afterwards it will not reset to 0 but stay at its last value. Only at the next counter step, will the first counter step be sent to the bus. Thus the counter will never have the value “0”.	
Additional object to store last value of counter on reset	**No** Yes Yes and send

No: no additional object to store the last value of the counter on reset will be activated. **Yes:** an additional object to store the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). **Yes and send:** an additional object to store and send the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). This value will then be sent after reset using this additional object.

A.2) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Run hour counter - BACK

Parameter	Settings
Data point type of counter	1 byte unsigned 2 bytes unsigned **4 bytes unsigned**
Usually, a Run hour counter has a 4 bytes unsigned value. But 1 and 2 bytes unsigned can also be configured for the purpose of showing the value in info displays, which cannot display 4 bytes unsigned values.	
Initial value run hour counter	**8000**
Attention! After programming this value will only be overwritten is the new starting value is changed.	

Here you can establish an initial value from which the counter will count back.

After downloading with the ETS this value will only be overwritten if the new starting value is changed.

Introduce here the lifespan of the connected load according to its data sheet which then can be used to supervise the lifespan of a lamp or any given load. It sends an alarm telegram when reaching the value zero. So instead of changing the lamp/load when it fails, it can be done before as a proactive measure. This is especially useful in halls with high ceilings. It cost more for a maintenance callout for changing individual bulbs every time they brake, than making a bulk replacement of all bulbs which or are close to or have reached zero, even though they are still working.

Should the conversion factor be activated and set to be for example “Several triggers decreases 1 step” = 3, and the “Initial value switching counter” is set to 5 then the sequence will be as follows: 444,333,222,111,000, and only at the last 0 the alarm will be sent.

Reaction on reaching zero	Stay at zero **Reset to initial value and start again**
Stay at zero: once the counter reaches 0, it will stay there until it has been reset. **Reset to initial value and start again:** once the counter reaches 0, it will start counting back again starting from the initial value of the run hour counter (as parameterized in the previous option).	
Additional functions	**No** Yes
In order to keep the application program as easy as possible, only the main and most important functions are displayed at first sight. You will often find the possibility to activate the Additional or Advanced Functions, which disclose new functions that are not essential, but can be very useful.	

a) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Run hour counter – BACK / ADDITIONAL FUNCTONS

Parameter	Settings
Cyclic sending of counter value	**No** Yes

When this function is activated, the corresponding object will not send the telegram once, but repeat it infinitely.	
Counter values are sent to the bus every: (Run hours)	1
Enter here the number of hours that must go by before the counter sends its value to the bus. This option is meant to reduce the bus traffic. For instance, if you enter a "5", the counter will have to count back 5 more hours in order to send the next value to the bus (60, 55, 50, 45, 40…).	
Conversion factor	**None** Several hours decreases 1 step 1 hour decreases several steps
None: for each 1 hour accumulated ON time of the channel, the counter decreases 1 step. **Several hours decrease 1 step:** define here the number of accumulated ON time (in hours) that must go by for the counter to decrease 1 step. **1 hour decrease several steps:** define here the step decrement for each hour of accumulated ON time. For example, after 8 accumulated ON time hours, the counter will have decreased 8 x 10 (= 80) steps.	
Send last value of counter at reset by counter object	**No** Yes
No: if you reset the counter by using the 1 bit reset object, the last value of the counter will not be sent to the bus by the counter object. Instead, a "0" will be sent to indicate it has been reset. **Yes:** if you reset the counter by using the 1 bit reset object, the counter object will send its current value before reset to the bus and afterwards it will not reset to 0 but stay at its last value. Only at the next counter step, will the first counter step be sent to the bus. Thus the counter will never have the value "0".	
Additional object to store last value of counter on reset	**No** Yes Yes and send
No: no additional object to store the last value of the counter on reset will be activated. **Yes:** an additional object to store the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse).	
Yes and send: an additional object to store and send the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). This value will then be sent after reset using this additional object.	

B) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Switching counter

Parameter	Settings
Switching counter	**No** Upward Backward
No: this option hides the Switching counter tab and all its objects and options. **Upward:** this option is used to count the accumulated switching operations of the current channel. **Backward:** to count down from a configurable initial value.	

B.1) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Switching counter - UP

Parameter	Settings
Data point type of counter	1 byte unsigned 2 bytes unsigned **4 bytes unsigned**
Usually, a Switching counter has a 4 bytes unsigned value. But 1 and 2 bytes unsigned can also be configured for the purpose of showing the value in info displays, which cannot display 4 bytes unsigned values.	
Count number of switching's on:	**Only ON** Only OFF ON and OFF

Only ON: the counter will increase only with ON operations.
Only OFF: the counter will increase only with OFF operations.
ON and OFF: the counter will increase with both ON and OFF operations.

Initial value switching counter	**No** Yes

Attention! After programming this value will only be overwritten is the new starting value is changed.

This option gives you the possibility to establish an initial value from which the counting will start up

After downloading with the ETS this value will only be overwritten if the new starting value is changed.

Practical example: should the actuator be installed in an existing installation, where the load connected to the current channel has already a known number of switching operations, this information can be used as the "New starting value". But in a later stage, if some other parameter in the actuator must be changed and downloaded, the new current counter value will not be overwritten.

Switching threshold value	0

Attention! 0 = Deactivated

Here you can enter the number of switching operations that will trigger the 1 bit alarm object of the current channel. So, this alarm object will be activated and send a "1" to the bus as soon as the switching counter passes this threshold.

Should the conversion factor be activated and set to be for example "Several switching's increases 1 step" = 3, and the threshold value is set to 5 then the sequence will be as follows: 0,0,1,1,1,2,2,2,3,3,3,4,4,4,5,… The alarm is sent in the first 5 after 15 pulses.

Attention, this alarm will also be sent to the bus immediately after bus recovery.

Object for reading / writing the threshold value	**No** Only readable Readable and writable

Only readable: this option will activate an unsigned counter object, which can be read by the ETS/other KNX devices.

- **Readable and writable:** this option will activate an unsigned counter object, which can be read and overwritten by the ETS/other KNX devices. This is meant to allow changing the threshold value with, for instance, a visualization.

Reaction on overflow (Max. value of DPT)	**Reset to 0 and start again** Stay at maximum

Attention! Both counter & alarm objects will be set to zero

Important note: the overflow must not be mistaken with the threshold value, since they are two totally different concepts:

- An overflow is reached when the object value exceeds the maximum value of the selected data point type. For example, the maximum value of a 1 byte unsigned value is 255; therefore, the overflow is reached when the object value exceeds 255.
- On the other hand, the threshold refers to any given value of your choice that is valid for this DPT.

Reset to 0 and start again: when then overflow is reached, the object will start counting from 0 again. Attention! In this case the alarm object will also be set to zero, otherwise one would not know if the threshold has newly been reached or not.
Stay at maximum: in the event of the overflow being reached, the object will stop at the maximum value of the DPT.

Additional functions	**No** Yes

In order to keep the application program as easy as possible, only the main and most important functions are displayed at first sight. You will often find the possibility to activate the Additional or Advanced Functions, which disclose new functions that are not essential, but can be very useful.

b) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Switching counter – UP / ADDITIONAL FUNCTONS

µBrick Actuator Series

Parameter	Settings
Cyclic sending of counter value	**No** Yes
When this function is activated, the corresponding object will not send the telegram once, but repeat it infinitely.	
Counter values are sent to the bus every: (Switchings)	1
Enter here the number of switching operations that be executed before the counter sends its value to the bus. This option is meant to reduce the bus traffic. For instance, if you enter a "50", the counter will send its first value whenever the accumulated switching operations of the channel amount to 50 and will then send the value 50 to the bus (50, 100, 150, 200, 250…).	
Conversion factor	**None** Several hours increases 1 step 1 hour increases several steps
None: for each switching operation of the channel, the counter increases 1 step. **Several hours increases 1 step:** define here the number of switching operations that must be executed for the counter to increase 1 step. **1 hour increases several steps:** define here the step increment for each switching operation. For example, after 50 switching operations, the counter will have increased 50 x 10 (= 500) steps.	
Send last value of counter at reset by counter object	**No** Yes
No: if you reset the counter by using the 1 bit reset object, the last value of the counter will not be sent to the bus by the counter object. Instead, a "0" will be sent to indicate it has been reset. **Yes:** if you reset the counter by using the 1 bit reset object, the counter object will send its current value before reset to the bus and afterwards it will not reset to 0 but stay at its last value. Only at the next counter step, will the first counter step be sent to the bus. Thus the counter will never have the value "0".	
Additional object to store last value of counter on reset	**No** Yes Yes and send
No: no additional object to store the last value of the counter on reset will be activated. **Yes:** an additional object to store the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). **Yes and send:** an additional object to store and send the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). This value will then be sent after reset using this additional object.	

B.2) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Switching counter - BACK

Parameter	Settings
Data point type of counter	1 byte unsigned 2 bytes unsigned **4 bytes unsigned**
Usually, a Run hour counter has a 4 bytes unsigned value. But 1 and 2 bytes unsigned can also be configured for the purpose of showing the value in info displays, which cannot display 4 bytes unsigned values.	
Count number of switching's on	**Only ON** Only OFF ON and OFF
Only ON: the counter will decrease only with ON operations. **Only OFF:** the counter will decrease only with OFF operations. **ON and OFF:** the counter will decrease with both ON and OFF operations.	
Initial value switching counter	**8000**
Attention! After programming this value will only be overwritten is the new starting value is changed.	

Here you can establish an initial value from which the counter will count back. Attention! This value will never be sent. The 1st value sent will be the first decreased value.

It will send a 1 bit alarm telegram with the value "1" when reaching the value zero.

After downloading with the ETS this value will only be overwritten if the new starting value is changed.

Introduce here the maximum number of switching's of the connected load,
(according to its data sheet) which then can be used to supervise the lifespan of a lamp or any given load. It sends an alarm telegram when reaching the value zero. So instead of changing the lamp/load when it fails, it can be done before as a proactive measure. This is especially useful in halls with high ceilings. It cost more for a maintenance callout for changing individual bulbs every time they brake, than making a bulk replacement of all bulbs which or are close to or have reached zero, even though they are still working.

Should the conversion factor be activated and set to be for example "Several triggers decreases 1 step" = 3, and the "Initial value switching counter" is set to 5 then the sequence will be as follows: 444,333,222,111,000, and only at the last 0 the alarm will be sent.

Reaction on reaching zero	Stay at zero **Reset to initial value and start again**

Stay at zero: once the counter reaches 0, it will stay there until it has been reset.
Reset to initial value and start again: once the counter reaches 0, it will start counting back again starting from the initial value of the switching counter (as parameterized in the previous option). Attention! This initial value will not be sent to the bus, the next trigger sends the decreased value.

Additional functions	**No** Yes

In order to keep the application program as easy as possible, only the main and most important functions are displayed at first sight. You will often find the possibility to activate the Additional or Advanced Functions, which disclose new functions that are not essential, but can be very useful.

b) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Switching counter – BACK / ADDITIONAL FUNCTONS

Parameter	Settings
Cyclic sending of counter value	**No** Yes
When this function is activated, the corresponding object will not send the telegram once, but repeat it infinitely.	
Counter values are sent to the bus every: (Switchings)	1
Enter here the number of switching operations that must be executed before the counter sends its value to the bus. This option is meant to reduce the bus traffic. For instance, if you enter a "50", the counter will have to count back 50 switching operations in order to send the next value to the bus (550, 500, 450, 400, 350…).	
Conversion factor	**None** Several hours decreases 1 step 1 hour decreases several steps
None: for each 1 switching operation of the channel, the counter decreases 1 step. **Several hours increases 1 step:** define here the number of switching operations that must be executed for the counter to decrease 1 step. **1 hour increases several steps:** de define here the step decrement for each switching operation. For example, after 50 switching operations, the counter will have decreased 50 x 10 (= 500) steps.	
Send last value of counter at reset by counter object	**No** Yes
No: if you reset the counter by using the 1 bit reset object, the last value of the counter will not be sent to the bus by the counter object. Instead, a "0" will be sent to indicate it has been reset. **Yes:** if you reset the counter by using the 1 bit reset object, the counter object will send its current value before reset to the bus and afterwards it will not reset to 0 but stay at its last value. Only at the next counter step, will the first counter step be sent to the bus. Thus the counter will never have the value "0".	
Additional object to store last value of counter on reset	**No** Yes Yes and send

µBrick Actuator Series

No: no additional object to store the last value of the counter on reset will be activated.
Yes: an additional object to store the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse).
Yes and send: an additional object to store and send the last value of the counter on reset will be activated. This object can work parallel with the previous option (Last value of counter at reset by counter object) and it is mainly there to store this last value until the next reset, whereas the counter object only stores it for a short time (until next counter pulse). This value will then be sent after reset using this additional object.

5.1.2.2 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Scenes

KNX standard 1 byte scenes: 1 Scene object per output. The advantage of having a Scene object per channel (and not only one for the all the channels) is that with the same Scene number, different scenes can be executed (since they are linked to another push button, with a different group address).
Up to 8 scenes can be configured per channel.

Parameter	Settings
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
Most of the actuator's modules can be deactivated with a "… disable" object. The value (1 or 0) used to disable can also be configured. This option can be very useful for many reasons, including simplifying the configuration: for instance, the logic functions might be a complex task that can take a while to finish; in the meantime, you don't want these modules to be active and cause unwanted actions. Therefore, you can disable them until you finish programming. Another example: you can simply activate/deactivate the timers for the irrigation system when not needed.	

5.1.2.2.1 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Scenes / COMMON SCENE PARAMETERS

As mentioned before, up to 8 scenes can be configured per channel with identical parameters.

Parameter	Settings
Reaction of channel for	Scene 1 … Scene 64
Attention! Same scene number may not be used twice! Only the first one (top) will prevail	
Here you can define the Scene number where this channel should participate in. All 64 possible KNX scenes can be used. As described in the KNX specifications, in order to reproduce scene 1, the value 0 has to be sent to the scene object of the channel and so on (0=play_scene1 …. 63= play_scene64). Important note: you may not use the same Scene number twice! Should you choose the same Scene number in more than one of the 8 available scene options, only the first one (from top to bottom) will prevail; the other will be ignored.	
Possible to save scene	No **Yes**
It is possible to save the current output state of the actuator as the new scene state. As described in the KNX specifications, in order to save scene 1, the value 128 has to be sent to the scene object of the channel and so on until 192 (128=save_scene1 …. 192= save_scene64). The configured parameter in "Output state for scene" will be overwritten. For example, the end user of the installation can switch ON/OFF the lights as wished and then save the current state for this scene via long press of a standard KNX scene push button. **No:** the scene cannot be saved with the KNX scene object. **Yes:** this option allows to overwrite the current state of the output as the new "Output state for scene", according to the KNX standardization. Important note: if the output state for scene is configured as a "Timer 1 reaction at ON" or "Timer 1 reaction at OFF", the output state will NOT be saved. The end-user parameters (like this one) can be configured in GENERAL SETTINGS/OVERWRITE END-USER PARAMETER VALUES AT DOWNLOAD. Here you can choose for the "Output state for scene" not to be overwritten by ETS download.	
Output state for scene	No function ON OFF Timer 1 reaction at ON Timer 1 reaction at OFF
Here you can establish the initial channel state of the scene. Please, note that this can be overwritten by the end user if you have selected "Yes" in the option above ("Possible to save scene"). **No function**: the channel will have no reaction in the initial stage; the channel will only react to this scene if "save scene" is active and it has been saved by the scene object. **ON:** the channel switches ON when executing the scene (unless otherwise saved via channel scene object) **OFF:** the channel switches OFF when executing the scene (unless otherwise saved via channel scene object)	

µBrick Actuator Series

Timer 1 reaction at ON: the function that has been chosen under “OUTPUTS/Timer 1/REACTION AT ON” will be executed (unless otherwise saved via channel scene object)
Timer 1 reaction at OFF: the function that has been chosen under “OUTPUTS/Timer 1/REACTION AT OFF” will be executed (unless otherwise saved via channel scene object)

5.1.2.3 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2

There are two timers linked to the current channel and which can run parallel; also, they have their own triggering object each. These timers can be configured to works as ON and/or OFF Delay, Staircase, Delay and staircase, blinking, etc.

The Timer trigger object is a 1 bit object which will have different behaviours when receiving an ON or OFF respectively. Next we will explain both REACTION AT ON and REACTION AT OFF separately:

5.1.2.3.1 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT ON

Parameter	Settings
REACTION AT ON	No action Delay **Staircase** Delay and staircase Only ON (without delay/staircase)
The timer can be used as any of the above timer types. These are the possible actions to be executed when the timer trigger object receives an ON ("1"): **No action:** the timer will not be executed. **Delay:** the channel switches ON after a time delay. **Staircase:** the channel immediately switches ON and stays ON for the configured staircase time and thereafter switches OFF again. **Delay and staircase:** the channel switches ON after a time delay and then stays ON for the configured staircase time and thereafter switches OFF again. **Only ON (without delay/staircase):** the channel immediately switches ON and stays ON.	

A) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT ON / Delay

Parameter	Settings
- ON delay Base	1 s
- ON delay Factor	10
Configure here the time delay for the channel to switch ON	

B) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT ON / Staircase

Parameter	Settings
- Staircase time (ON duration) Base	**1 s** 5 s 10 s 1 min 5 min 10 min 1 h
- Staircase time (ON duration) Factor	**60**
Establish here the wished time for the channel to be ON The Staircase time is the period of time during which the actuator channel will be switched ON. After this time elapses, the channel switches OFF again.	
Staircase time Factor changeable by object	**No** Yes
No (default option): staircase time only configurable via parameters. Yes: this option activates an object to change staircase time factor. As you can see in the picture below, the time Base can be any of the following: So, if you have selected, for instance, "1 s", then the values received in this object will be in "seconds". If you have selected "5 s" though, the values received will be in "seconds" and multiplied by 5 (base "5 s" x value received at object "10" = "50 seconds"). The same rule applies if the Base has been selected in "minutes" or "hours". When using this communication object to modify the staircase factor, if the modification is done while the staircase is active , the modification will be applied after the end of the current staircase	

Advanced staircase function	**No** Yes
Here the advanced functions can be activated.	

C) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT ON / ADVANCED STAIRCASE FUNCTIONS

Parameter	Settings
Multiply staircase	**No** Yes

* With Yes: ***Attention! Total staircase time = staircase time x number of consecutive ON telegrams separated by less than 1 sec. from each other***

Here you can activate the possibility to multiply the staircase time in order to extend the time during which the channel will stay ON. The total staircase ON time is calculated by taking the parameterized staircase time and multiplying it by the number of consecutive ON telegrams received.

Keep in mind that the multiplication telegrams (consecutive ON telegrams) must be separated by less than 1 second from each other. Should more than 1 second elapse between two telegrams, then it will only do the multiplication of the previous pulses received. The telegrams received after this, will be ignored or interpreted as a retrigger timer function (if parameterized), see next parameter "Retrigger timer" option.

This resulting multiplication time will never exceed the maximum staircase time as can be configured in the parameter option "Maximum staircase time Base/Factor"

Practical example: as implied by its name, the staircase time is frequently used in staircases. With the purpose of lowering the costs, instead of using a movement detector for switching ON/OFF, often push buttons are used with the staircase time as defined in the actuator. In order to save energy, the staircase time should be as short as possible, but sometimes you may wish to have the lights longer ON. In this case, this option can be very useful because it allows the end user to easily extend the staircase time by pressing several times (depending on how long the light should stay ON).

Retrigger timer	No **Yes, excluding multiplication** Yes, including multiplication

It is possible to extend the staircase time by retriggering it (in other words, the timer starts counting again from the start) with an ON telegram. But this function will only be executed after more than 1 second has elapsed between the triggering events of the timer (for trigger events less than 1 second, see the behaviour in the section "MULTIPLY STAIRCASE").

Keep in mind that only the "Staircase time (ON duration)" will be extended. (So if the staircase is configured with an ON delay, when receiving the retrigger telegram it will NOT switch OFF, and the ON delay will be ignored)

If the previous parameter option "Multiply staircase" is activated, the retrigger telegrams will also do the multiplication, given the consecutive ON telegrams are separated by less than 1 second from each other.

No: the staircase will not be retriggered.

Yes, excluding multiplication (default option): this option will retrigger the staircase to be reset to the time (Base/Factor) as configured in the ETS application program.

For example: you have configure the staircase time in the ETS application program to be 1 minute; should the staircase time be, for instance, 1 hour as the result of a previous multiplication (Multiply staircase option), the moment you receive the retrigger telegram it will be reset to 1 minute again.

Yes, including multiplication: this option will retrigger the staircase to be reset to the current staircase time (it could be the parameterized time or the multiplied staircase time).

For example: you have configure the staircase time in the ETS application program to be 1 minute; should the staircase time be, for instance, 1 hour as the result of a previous multiplication (Multiply staircase option), the moment you receive the retrigger telegram it will be reset to 1 hour again.

Warning pulse	**No function** With own output With additional object

The warning pulse is meant to inform the end user about the fact that the staircase time is about to expire.

No function: the light will go OFF without previous warning after the staircase time elapses.

With own output: the same channel will be used for this warning pulse.

According to the default parameters, the output will switch OFF 10 seconds before the end of the staircase time and it will switch ON again 2 seconds after switching OFF. This creates a short blinking effect as a visual warning.

It is important to be able to configure the OFF time because not all loads can switch OFF immediately (for example, lights using transformers). So, if you have selected 1 second as a warning time, it might not switch OFF at all.

With additional object: this option serves the same purpose of warning before the staircase time elapses. It is specially indicated for those places where the channel can/may not be switched ON and OFF quickly. In these cases, the additional object can send a warning pulse to another channel (different load) just before the end of the staircase time of the main load.

Practical example: let's say this channel is used to control the flood lights of a tennis court via contactor. These lights take long to switch ON again (after they have been switched OFF), which is not energy-efficient nor practical. Therefore, to be able to generate a warning pulse, you can use an additional warning light connected to another channel, which this additional object is linked to.

1 action: ON: the additional object only sends a "1" at the configured point in time before the staircase time elapses.

2 actions: 1st OFF, 2nd ON: the additional object can execute two actions by sending:
- Time before end of staircase for 1st action: a "0" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 2nd action: a "1" at the configured point in time before the staircase time elapses.

2 actions : 1st ON, 2nd OFF: the additional object can execute two actions by sending:

- Time before end of staircase for 1st action: a "1" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 2nd action: a "0" at the configured point in time before the staircase time elapses.

3 actions: 1st OFF, 2nd ON, 3rd OFF (default option): the additional object can execute three actions by sending:
- Time before end of staircase for 1st action: a "0" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 2nd action: a "1" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 3rd action: a "0" at the configured point in time before the staircase time elapses.

D) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT ON / Delay and staircase

The Staircase function has been explained above. This "Delay and Staircase" combined function could also have:

Parameter	Settings
- ON delay Base	**1 s**
- ON delay Factor	**10 s**
The staircase can start after a configurable time delay	
- Staircase time (ON duration) Base	**1 s**
- Staircase time (ON duration) Factor	**60 s**
Establish here the wished time for the channel to be ON The Staircase time is the period of time during which the actuator channel will be switched ON. After this time elapses, the channel switches OFF again.	
Staircase time factor changeable by object	**No** Yes

No (default option): staircase time only configurable via parameters.

Yes: this option activates an object to change staircase time factor. As you can see in the picture below, the time Base can be any of the following:

So, if you have selected, for instance, “1 s”, then the values received in this object will be in “seconds”. If you have selected “5 s” though, the values received will be in “seconds” and multiplied by 5 (base “5 s” x value received at object “10” = “50 seconds”). The same rule applies if the Base has been selected in “minutes” or “hours”.

Blinking / number of repetitions (0 = none, 65535 = infinite)	**0**

A repeated staircase function with an initial delay actually becomes a blinking function. It is indicated to switch a load ON and OFF with a configurable certain frequency (which can have different ON and OFF times).

The number of repetitions can be configured and can also be set to any number between 1 and 65534.Infinite repetitions can be achieved by using the value 65535.

In order to deactivate the blinking, just enter the value 0.

5.1.2.3.2 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 /REACTON AT OFF

Parameter	Settings
REACTION AT OFF	No action **OFF without delay** OFF with delay

Attention! Reaction at OFF cancels the running staircase

This are the possible actions to be executed when the timer trigger object receives an OFF (“0”):

No action: the timer will not be interrupted.

OFF without delay: the channel immediately switches OFF and the timer function is cancelled.

OFF with delay: the channel switches OFF after a time delay.

As soon as the OFF telegram is received, the Timer is cancelled.

Object to disable timer	Yes, immediately Yes, on ending current timer **No**

The disable object will always react as follows (and cannot be otherwise configured):

“1”: disable.
“0”: enable.

Yes, immediately: as soon as the Disable object receives a “1”, the timer will be cancelled and disabled. This option activates the parameter “Reaction on bus voltage recovery”.

Yes, on ending current timer: whenever the Disable object receives a “1”, the timer will be not cancelled, but disabled. Thus, the current timer will finalize normally. This option activates the parameter “Reaction on bus voltage recovery”.

No: the disable object, including the “Reaction on bus voltage recovery” will be hidden.

A) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT OFF / Object to disable timer

Parameter	Settings
Objecto to disable timer	**Yes, immediately** Yes, on ending current timer No

The disable object will always react as follows (and cannot be otherwise configured):
- “1”: disable.
- “0”: enable.

Yes, immediately: as soon as the Disable object receives a “1”, the timer will be cancelled and disabled. This option activates the parameter “Reaction on bus voltage recovery”.
Yes, on ending current timer: whenever the Disable object receives a “1”, the timer will be not cancelled, but disabled. Thus, the current timer will finalize normally. This option activates the parameter “Reaction on bus voltage recovery”.
No: the disable object, including the “Reaction on bus voltage recovery” will be hidden.

µBrick Actuator Series

A.1) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT OFF / Object to disable timer / Reaction on bus voltage recovery

Parameter	Settings
Reaction on bus voltage recovery	**Enable** Disable Last object status
Whether the Timer will be active or not on bus voltage recovery can be configured here. On bus voltage recovery the timer can be enabled, disabled, or have the same state as before the bus failure depending on the above selection. **Enable:** the timer will be enabled. **Disable:** the timer will be disabled. **Last object status:** the status of the Enable object will be saved in the actuator's non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will set the object as it was before the bus failure.	

B) Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Timer 1 and 2 / REACTION AT OFF / Reaction when SWITCHING or SCENE objects receive a value while timer is active

Parameter	Settings
Reaction when SWITCHING or SCENE objects receive a value while timer is active	Don't cancel timer and do action **Cancel timer and do action** Ignore telegram
Don't cancel timer and do action: the Switching or Scene function will not cancel the active timer and the function will be executed parallel to the Timer. **Cancel timer and do action:** the Switching or Scene function will cancel the active timer and only the triggered functions (Switching or Scene) will be executed (whereas the Timer will be cancelled and thus will not interfere with these functions). **Ignore telegram:** if a telegram is received via the Switching or Scene objects while the timer is active, these functions (Switching or Scene) will not be executed.	

5.1.2.4 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Disable

Each and every channel has a Disable object, which blocks all other functions of the channel.

The behaviour at Disabling/Enabling can be configured per channel.

On the other hand, the priority of all Disable objects can also be adjusted to have higher/lower priority as the alarms; this can be done in General Settings/Advanced Functions/Alarms (then, Alarm tab)

Parameter	Settings
Disable object	**Disable with ON** Disable with OFF
Disable with ON: the channel will be blocked whenever the Disable object receives a "1"; and enabled again with a "0". **Disable with OFF:** the channel will be blocked whenever the Disable object receives a "0"; and enabled again with a "1".	
- Reaction on bus voltage recovery	**Enable** Disable Last object status
Whether the channel will be disabled or enabled on bus voltage recovery can be configured here. **Enable:** the channel will be enabled. **Disable:** the channel will be disabled. **Last object status:** the status of the Enable object will be saved in the actuator's non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will set the object as it was before the bus failure.	
Behaviour at disabling	**Block channel as is** ON OFF Timer 1 reaction at ON Timer 1 reaction at OFF
Block channel as is: the channel will be blocked, but not switched ON or OFF when disabling the channel via Disable object. **ON:** the channel will be switched ON and blocked. **OFF:** the channel will be switched OFF and blocked. Each output has two timer functions. Only the first timer can be assigned to the behaviour at disabling: **Timer 1 reaction at ON:** the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT ON" will be executed and the channel will be blocked. **Timer 1 reaction at OFF:** the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT OFF" will be executed and the channel will be blocked.	
Behaviour at enabling	**Enable and leave channel as is** ON OFF Timer 1 reaction at ON Timer 1 reaction at OFF Set to tracked state
Enable and leave channel as is: the channel will be enabled, but not switched ON or OFF when enabling the channel via Disable object. **ON:** the channel will be switched ON and enabled. **OFF:** the channel will be switched OFF and enabled. Each output has two timer functions. Only the first timer can be assigned to the behaviour at enabling: **Timer 1 reaction at ON:** the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT ON" will be executed and the channel will be enabled. **Timer 1 reaction at OFF:** the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT OFF" will be executed and the channel will be enabled. **Set to tracked state:** while the channel is blocked, the other channel-related objects might receive telegrams. Nevertheless, since the channel is blocked, it does not switch ON or OFF. Even though the actuator does not switch ON or OFF, it does register all these events in order to be able to go to the state where it would have been at enabling (if the channel had not been blocked).	
Attention! Enable channel will trigger the behaviour of the next active (lower priority) alarm. Also the "Behaviour at enabling" will only be executed with no active & acknowledged channel alarms.	

5.1.2.5 Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Alarms

Attention! Alarm function must be activated in "General Settings" tab

First of all, in order for the channel-related Alarms to work, the Alarms must be activated in "General Settings/Advanced Functions/Alarms". In this tab you can configure up to 8 alarms to be either "analogue" or "digital".

Channel-dependent alarms: now, in the Advanced Functions of the current channel, you can configure the behaviour of the channel when the alarm objects receive a telegram.

After choosing the "Yes" option, the channel-related Alarms tab will be displayed.

Alarm telegrams are used to block the channel. The reaction of the current channel when any/several of the 8 available alarms have been activated can be configured in the next tab.

Parameter	Settings
Behaviour at beginning of alarm 1…8	**Nothing** Block channel as is ON OFF Timer 1 reaction at ON Timer 1 reaction at OFF

Nothing: the channel will not participate in the alarm. Thus, it will not be blocked.
Block channel as is: the channel will be blocked, but not switched ON or OFF when activating the alarm.
ON: the channel will be switched ON and blocked.
OFF: the channel will be switched OFF and blocked.

Each output has two timer functions. Only the first timer can be assigned to the behaviour of the alarm:
Timer 1 reaction at ON: the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT ON" will be executed and the channel will be blocked.
Timer 1 reaction at OFF: the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT OFF" will be executed and the channel will be blocked.

Behaviour at end of all alarms	**Nothing** ON OFF Timer 1 reaction at ON Timer 1 reaction at OFF Set to tracked state

Attention! The "Behaviour at end of all alarms" will only be executed with no active & acknowledged channel alarms, and if the "disable channel function" is in enabled state. Only then, the channel will be unblocked.

Here you can define the behaviour of the current channel when no alarm is active anymore.

Important note: in the General Settings tab you can configure whether or not the alarms must be acknowledged. The "Behaviour at end of all alarms" will only be executed with no active & acknowledged channel alarms, and if the "disable channel function" is in enabled state. Only then, the channel will be unblocked.

Nothing: the channel will not do anything when enabled.
ON: the channel will be switched ON when enabled.
OFF: the channel will be switched OFF when enabled.

Each output has two timer functions. Only the first timer can be assigned to the behaviour at enabling:
Timer 1 reaction at ON: the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT ON" will be executed when enabled.
Timer 1 reaction at OFF: the function that has been chosen under "OUTPUTS/Timer 1/REACTION AT OFF" will be executed when enabled.

Set to tracked state: while the channel is blocked, the other channel-related objects might receive telegrams. Nevertheless, since the channel is blocked, it does not switch ON or OFF.

Even though the actuator does not switch ON or OFF, it does register all these events in order to be able to go to the state where it would have been at enabling (if the channel had not been blocked).

5.2 Parameter page: OUTPUTS / Channel X1 (Shutter / blind)

One channel can be used as either two separate relay outputs or as one Shutter / Blind channel. When selecting blind/shutter, the outputs will be interlocked with each other. Meaning that only one output relay can be closed at a time. In order to close one of the channels the other must first be opened.

With these two outputs the blind can be moved (up/down or to a specific position). The channel must always know its current position and therefore it must sometimes be calibrated.

The blind will always be calibrated on the first movement after an ETS download. This calibration procedure can always be interrupted by sending any movement or stop telegram to the channel.

Please, see OUTPUT: CHANNEL TYPE SELECTION and OUTPUT TYPE SELECTION before proceeding.

1 bit Move object	Value received = 0	UP movement
	Value received = 1	DOWN movement
Absolute position shutter/blind	Totally UP	0%
	Totally DOWN	100%
Absolute position slat	Totally UP	0%
	Totally OPEN	50% (usually)
	Totally DOWN	100%

SHUTTER TABLE: KNX standard specifications for shutter/blinds

After choosing “Shutter / Blind”, the following two tabs will be automatically activated, as well as the relevant Shutter objects.

1.- Shutter tab for the current Channel: in this tab you must select the type of drive connected to the channel.
2.- Shutter Status tab for the current Channel

Parameter page: OUTPUTS / Channel X1 (Shutter / blind)

Parameter	Settings
Type	**Shutter (without slats)** Blind (with slats)

Attention! All slats parameters will be ignored

Important note “Shutters”: due to ETS technical characteristics, it is not practical to hide all non-applicable, slat related options in the Shutter drop down context menus. So, when you select “Shutter (without slats)”, please ignore the slats parameters (if you select any slat parameter while configuring shutters, these will have no effect at all).

By working this way, the common objects and the assigned group addresses will not be deleted when changing from shutters to blinds or vice versa. This could be a great advantage, should the final user change the elements of the installation at any point in time.

Important note “Blinds”: if you select “Blinds (with slats)”, all Shutter parameters still apply identically (only Status tab is a totally new one). Furthermore, you will find these additional functions:

- The “SLATS PARAMETERS” general configuration menu.
- Also the additional slats options will be now applicable in the Shutter drop down context menus.

In this manual, those additional parameters that apply only to slats (blinds) configuration, will appear in brown colour.

Travel time movement UP	**1 s**

This is the period of time during which the current Channel’s UP (first) relay will be closed and then opened again for a full movement (from 100% to 0%).

To calculate the total Travel Time of a blind (with slats) you must ignore the period of time while the slats are changing. Only the time while the blind is moving UP/DOWN must be counted

Different travel time for movement DOWN	**No** Yes

Sometimes (especially when controlling heavy shutters) the shutter moves much faster DOWN than UP. Here you can parameterize the travel time for a full DOWN movement (from 0% to 100%).

This is important for the actuator to be able to calculate the absolute position (0-100%) correctly.

Time for direction change	**500 ms**

This is the time that must go by while moving in one direction to change to the opposite direction.

For instance, if you receive a movement DOWN while the shutter is moving UP (first relay of the channel is closed), then the first relay must open and the second relay must close in order to move the blind DOWN. The time for closing the second relay (after opening the first relay) is configured here.

This time must be, at least, 500ms, since the two relays for the Shutter output may never be closed at the same time.

Practical tip: due to the inertia of heavy shutters, you must be able to extend this time in order to give the shutter the chance to stop before changing direction.

5.2.1 Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS

This functionality only appears when you have chosen "Blinds (with slats)".

Parameter	Settings
Total slat time from 0 to 100%	100 ms 500 ms **1 s** 10 s 1 min 10 min 1 h
Attention! This time should be longer than time for long oper, in push button	

Here you can configure (unlike with many other blinds actuators in the market) not the time for each slat movement, but the total time for a slat to execute a full movement from 0 to 100%.

The reason for this is the fact that the slat movement steps are very short and are difficult to calculate. Also, usually it is more practical to configure the NUMBER OF SLATS STEPS to complete a full movement (than calculating each step time).

Note: the time you choose here should be longer than that used for the long press of a standard KNX shutter/blind push button. Otherwise, the blind will have an undesired behaviour as in the following sequence:

1. MOVE: By pressing the button (most push buttons immediately send the first telegram), the blind will immediately start to move during the time configured here.
2. STOP: So, because this time is shorter, the blind will stop before the time for long operation in the push button has elapsed.
3. MOVE AGAIN: Then, since you are still pressing the button when the time for long operation in the push button has been reached, the blind will start moving UP/DOWN (for the configured total blind time).

Number of slats steps	**5**
Here you can configure the number of steps to be made in a full slat movement from 0 to 100%.	
Maintain slat position after blind movement	No **Yes**
When this option has been selected (as it is by default), the slats will automatically return to the position they were in before the UP/DOWN movement. Take into account that the next parameter option "Slat position after reaching bottom …" has priority over this parameter and if it is selected, the previous slat position will not be maintained.	
Slat position after reaching bottom position % (100%=disabled)	**100**

Here you can enter the position the slat must move to after a full movement DOWN (100%). This option can be disabled by entering the value 100 (%). Also note that it has preference over "Maintain slat position after blind movement".	
Bus failure	**No** Yes
No: this option hides the Bus failure tab and all its functions. If the blind is moving when the bus fails it will stop (open both relays) immediately and it will store this position in the non-volatile memory. Therefore on bus voltage recovery no calibration movement is needed. **Yes:** this option opens the Bus failure tab, which allows the configuration of the reaction of the channel on bus voltage failure/recovery.	
Advanced functions	**No** Yes
The µBrick Actuator range is also a powerful controller module (logic, timer, counter, etc. module). You can find Advanced Functions: 3) In the General Settings parameter page: this a totally independent controller module, with its own input and output objects, which can work autonomously (no need to be linked to any actuator function). On top of that, the most common advanced functions are also available within each and every channel. The main difference is that these are linked to the channel and cannot be used independent from it. This has the advantage that it is not necessary to use group addresses to link them, making configuration easier.	
Manual control	No **Yes**
Attention! Manual control must be activated in outputs	
The µBrick actuator has 2 push buttons and status LEDs on the front side. These buttons can be used to control the current channel if you select "yes" in this parameter option. Please, see **Annex 1** to learn more about manual control.	

5.2.1.1 Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Bus failure

Parameter	Settings
Reaction on bus voltage failure	Unchanged Up Down **Stop**
Attention! When selecting "Up" or "Down", the relay will close and stay closed. In case of direction change it will be almost immediate ("Time for direction change" cannot be executed).	
Unchanged: whenever the bus voltage fails, the contact stays the same. **Up:** whenever the bus voltage fails, the first relay will be opened and the second closed. **Down:** whenever the bus voltage fails, the second relay will be opened and the first closed. Important note for UP/DOWN: since the actuator only has a short time buffer to do the actions on bus voltage failure, it cannot open the relay again after UP/DOWN movement. Therefore, the relay will stay in the same position until bus voltage recovery (depending on the Bus voltage recovery configuration). This can be dangerous because the relay will be permanently closed and could still be under tension. If the bus fails while the blind was moving and if this parameter "Reaction on bus voltage failure" is set to either "Unchanged", "Up" or "Down" the blind will make a calibration movement on the next telegram received to move the blind. In this case it will also do a calibration movement if the next parameter "Reaction on bus voltage recovery" is set to "Position", "Move to slat and blind position", "Preset" or "Recovery status before bus failure" as soon as the bus recovers. **Stop:** whenever the bus voltage fails, both contacts open. With this option selected the blind will not do a calibration movement when bus voltage returns nor when receiving a telegram to move the blind.	
Reaction on bus voltage recovery	**Stop** Up Down Position Move to slat and blind position Preset Recovery status before bus failure
Stop: whenever the bus voltage returns, both contacts open. **Up:** whenever the bus voltage returns, the channel moves UP. The second relay will be opened; and the first relay will	

be closed for the full “Travel time movement UP”, independent of the current blind position.
Down: whenever the bus voltage returns, the channel moves DOWN. The first relay will be opened; and the second relay will be closed for the full “Travel time movement UP”, independent of the current blind position. If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN.
Position: whenever the bus voltage returns, the shutter will move to a certain position (0-100%), which can be parameterized here.
Move to slat and blind position: not applicable for shutter configuration.

Blinds (with slats): whenever the bus voltage returns, the blind and the slats will move to a certain position (0-100%)

Preset: you can select one of the four previously configured PRESETS (Channel/Advanced Functions) to be executed on bus voltage recovery.
Attention! Presets parameters must be configured in Channel -> Advanced functions
Recovery status before bus failure: the status of the output will be saved in the actuator’s non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will move the shutter to the position previous to the bus failure.

> **Important note on calibration**: for “Position”, “Move to slat and blind position”, “Preset” and “Recovery status before bus failure”.
>
> ***Attention! An absolute position on bus power recovery will cause a calibration movement to the upper end position***
>
> Sometimes it is impossible for the actuator to know the exact position of the shutter: for instance, on bus voltage return (the power failure of the bus and that of the current shutter are independent from each other) or with heavy shutters having made several absolute position movements (without having reached the end position).
>
> In these cases, the actuator needs to calibrate itself by making a full movement to the 0/100% position (upper/lower end position) before moving to the desired absolute position.
>
> After calibration, the shutter now has a reference from where to part again for the next movement.

5.2.1.2 Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Advanced functions

Parameter	Settings
Scenes	**No** Yes
KNX standard 1 byte scenes: 1 Scene object per output. The advantage of having a Scene object per channel (and not only one for the all the channels) is that with the same Scene number, different scenes can be executed (since they are linked to another push button, with a different group address). Up to 8 scenes can be configured per channel. **No:** this option hides the Scenes tab and all scene related functions and object for the current channel. **Yes:** this option activates the Scene tab, with the following functions and the Scene object for this channel. Important note: please see END-USER PARAMETERS	
Presets	**No** Yes
Presets are fixed absolute-positions of the shutter which are executed with a 1 bit object to move the shutter to a specific position. KNX Scenes are always executed with the 1 byte KNX scene object. But sometimes you might want to set the shutter to a specific position with, for instance, a central ON/OFF 1 bit command. In these cases, you can use a Preset, instead of a scene. **No:** this option hides the preset tab and related objects. **Yes:** this option activates the preset tab and, by default, also the first preset and its object.	
Alarms	**No** Yes
Attention! Alarm function must be activated in "General Settings" tab	
First of all, in order for the channel-related Alarms to work, the Alarms must be activated in General Settings/Advanced Functions/Alarms. In this tab you can configure up to 8 alarms to be either "analogue" or "digital". CHANNEL-DEPENDENT ALARMS Now, in the Advanced Functions of the current channel, you can configure the behaviour of the channel when the alarm objects receive a telegram. After choosing the "Yes" option, the channel-related Alarms tab will be displayed. Alarm telegrams are used to block the channel. The reaction of the current channel when any/several of the 8 available alarms have been activated can be configured in the next tab.	
Disable	**No** Yes
Apart from the Alarms, this is another way to block the channel. The main difference is that there is a Disable object for each channel, whereas the Alarm objects are common objects (for all assigned channels). **No:** this option hides this functionality and its related object. **Yes:** this option activates the Disable tab.	
Inverted movement object	**No** Yes
No: this option hides the "Move inverted" object. **Yes:** this option activates the so called "Move inverted" object, which is an additional object to the normal "Move" object. As you can see in the Shuter table, the shutter usually moves down with a "1" and up with a "0". With this object you can invert those values.	
Central UP/DOWN function	**No reaction** Any value = Up Any value = Down Any value = Position 0 = Up, 1 = Down 1 = Up, 0 = Down 0 = X, 1 = Down 0 = Up, 1 = X
Attention! Alarm function must be activated in "General Settings" tab	
In order to do a classic KNX "Central function", this actuator has a specific option that allows all the channel actions at once with only one or two objects. This considerably reduces the amount of group address associations (both meant to ease programmers work load, but also to reduce the actuator's association table). Before we configure the function within the channel, we must go to GENERAL SETTINGS / CENTRAL ON/OFF, UP/DOWN OBJECT and activate one of the objects.	

µBrick Actuator Series

The actuator has 1 or 2 Central ON/OFF, UP/DOWN objects for binary outputs and/or shutter (depending on the configuration in "General Settings/Outputs"):

- o 1 common object = "Central switching/move blind"
- o 2 separate objects = "Central switching" + "Central move"

No reaction: the channel has no reaction when the Central UP/DOWN object/s receive/s a telegram.
Any value = Up: the channel moves UP when the Central UP/DOWN object/s receive/s any telegram (no matter whether "0" or "1" is received).
Any value = Down: the channel moves DOWN when the Central UP/DOWN object/s receive/s any telegram (no matter whether "0" or "1" is received).
Any value = Position: the channel moves to a certain position when the Central UP/DOWN object/s receive/s any telegram (no matter whether "0" or "1" is received).
0 = Up, 1 = Down: the channel moves UP when the Central UP/DOWN object/s receive/s a "0" and moves DOWN when receiving a "1".
1 = Up, 0 = Down: the channel moves UP when the Central UP/DOWN object/s receive/s a "1" and moves DOWN when receiving a "0".
0 = X, 1 = Down: the channel has no reaction when the Central UP/DOWN object/s receive/s a "0" and moves DOWN when receiving a "1".
0 = Up, 1 = X: the channel moves UP when the Central UP/DOWN object/s receive/s a "0" and has no reaction when receiving a "1".

Limit travelling range / Manual calibration	

Attention! upper limit must be smaller than lower limit, otherwise it will be ignored

Attention! Calibration forces movement to end position, even if limits have been set

With this option you can change both the limits maximum and minimum end positions. The upper limit must be smaller than the lower limit, otherwise it will be ignored.

No: the blind moves from 0-100%.
With "No", the option "Additional time (after reaching end position" appears:

This is the additional time (in seconds) after having reached one of the end positions (0-100%) during which the output will still be closed in order to make sure that the end position has been reached. When the blind is in 0% and a up command is received the blind will move up during this "Additional time…". The same will happen when receiving a command to move down while the blind is at 100%.

Due to the mechanical friction of the shutter, which is not identical in each movement, the time to move the shutter UP/DOWN might sometimes be longer than the previously measured shutter time. This fact can cause that the shutter never reaches the end position (top/bottom) as expected. By using this additional time, the relay will stay closed for this period of time even though the actuator might have already reached 0-100%, thus ensuring that the end position is reached in any case.

Parameters: here you can adjust the upper and lower limits of the shutter's course of movement. This option will also activate a 1 bit object which can be used to disable the limits and enable them while forcing a calibration movement. Disable = 0 / Enable and calibrate = 1

Practical tip: should no limits be needed, this function could be used to manually calibrate the blinds by setting the upper limit to 0% and the lower limit to 100% and to send a 0 followed by 1 to the "Disable limits / calibrate" object.

Via two 1 byte objects: the two 1 byte scaling (0-100%) objects "Change upper limit" and "Change lower limit" are activated. They can be used to set the shutter's maximum and minimum end-position. If you send an invalid value (upper limit > lower limit or vice versa) to any of the limit objects, this value will be discarded and the object will resend the previous value to the bus. This way the user will note that this value was invalid.

This option will also activate a 1 bit object which can be used to disable the limits and enable them while forcing a calibration movement. Disable = 0 / Enable and calibrate = 1

Both: this option activates both the Parameters and the 1 byte objects. The goal is to have initial limits that can be changed in a later stage.

µBrick Actuator Series

Calibrate blinds outputs by moving to end position	**No** Shortest way Upper end position Lower end position
Sometimes the current blind position and the actuators status blind position get out of sync, especially with heavy shutters having made several absolute position movements (without having reached the end position). In these cases, the actuator needs to calibrate itself by making a full movement to the 0/100% position (upper/lower end position) before moving to the desired absolute position. After calibration, the shutter now has a reference from where to part again for the next movement. **No:** no calibration will be executed. **Shortest way:** the actuator calculates the shortest distance to the end position and makes a full movement of the shutter in that direction to ensure that the end position has been reached. **Upper end position:** the shutter makes a full movement UP (the first relay will be closed during the configured TRAVEL TIME MOVEMENT UP) to ensure that the end position has been reached. **Lower end position:** the shutter makes a full movement DOWN (the second relay will be closed during the configured TRAVEL TIME MOVEMENT UP. If a different time has been defined for moving down, then the time will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN) to ensure that the end position has been reached.	
Manual control	No **Yes**
Attention! Manual control must be activated in outputs	
The µBrick actuator has 2 push buttons and status LEDs on the front side. These buttons can be used to control the current channel if you select “yes” in this parameter option. You can see the exact behaviour of these buttons in OUTPUTS / MANUAL CONTROL.	

µBrick Actuator Series

A) Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Advanced functions / Scenes

A.1) Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Advanced functions / Scenes / Enable / Disable object

Parameter	Settings
Attention! The end-user parameter values will only be maintained when "overwrite end-user…" in general tab were set to "Don't overwrite". Important note: please see END-USER PARAMETERS	
Enable / Disable objects	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
Most of the actuator's modules can be deactivated with a "… disable" object. The value (1 or 0) used to disable can also be configured. This option can be very useful for many reasons, including simplifying the configuration: for instance, the logic functions might be a complex task that can take a while to finish; in the meantime, you don't want these modules to be active and cause unwanted actions. Therefore, you can disable them until you finish programming. Another example: you can simply activate/deactivate the timers for the irrigation system when not needed.	

A.2) Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Advanced functions / Scenes / Common scene parameters

As mentioned before, up to 8 scenes can be configured per channel with identical parameters.

Parameter	Settings
Attention! Same scene number may not be used twice! Only the first one (top) will prevail	
Important note: you may not use the same Scene number twice! Should you choose the same Scene number in more than one of the 8 available scene options, only the first one (from top to bottom) will prevail; the other will be ignored.	
Reaction of channel for	**Scene 1** … Scene 64
Here you can define the Scene number where this channel should participate in. All 64 possible KNX scenes can be used. As described in the KNX specifications, in order to reproduce scene 1, the value 0 has to be sent to the scene object of the channel and so on (0=play_scene1 …. 63= play_scene64).	
Output state for scene	**No function** Up Down Move to position Move to slat and blind position Move to preset
No function: the channel will have no reaction in the initial stage; the channel will only react to this scene (If "save scene" is active), and it has been saved by the scene object. **UP:** the channel moves UP when executing the scene (unless otherwise saved via channel scene object) **DOWN:** the channel moves DOWN when executing the scene (unless otherwise saved via channel scene object) **Move to position:** the shutter will move to a certain position (0-100%) when executing the scene (unless otherwise saved via channel scene object); the exact position can be parameterized here. **Move to slat and blind position:** not applicable for shutter configuration. Blinds (with slats): the blind and the slats will move to a certain position (0-100%), which can be parameterized here. **Move to preset:** the shutter will move to one of the four previously configured PRESETS (Channel/Advanced Functions) when executing the scene (unless otherwise saved via channel scene object).	
Possible to save scene	No **Yes**

It is possible to save the current position of the shutter as the new scene state.

As described in the KNX specifications, in order to save scene 1, the value 128 has to be sent to the scene object of the channel and so on until 192 (128=save_scene1 …. 192= save_scene64).

The configured parameter in OUTPUT STATE FOR SCENE will be overwritten. For example, the end user of the installation can move the shutter UP/DOWN as wished and then save the current position for this scene via long press of a standard KNX scene push button.

No: the scene cannot be saved with the KNX scene object.
Yes: this option allows to overwrite the current position of the shutter as the new OUTPUT STATE FOR SCENE, according to the KNX standardization.
Important note:
The END-USER PARAMETERS (like this one) can be configured in GENERAL SETTINGS/OVERWRITE END-USER PARAMETER VALUES AT DOWNLOAD. Here you can choose for the “Output state for scene” not to be overwritten by ETS download.

B) Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Advanced functions / Presets

Parameter	Settings
Attention! The end-user parameter values will only be maintained when "overwrite end-user…" in general tab were set to "Don't overwrite". Important note: please see END-USER PARAMETERS	
PRESET 1	**Yes** No
PRESET 2 … PRESET 4	Yes **No**
There are 4 Presets available (only the first of which is, by default, activated) Presets are predefined positions of the blind and or slat position which can be reproduced by sending a "1" to the object to execute the preset.	
Set initial default positions	No function **Only movement position** Only slat position Movement and slat position
No function: no preset position can be set as default value in the parameters; the 1 bit preset object is still available, though. In order to set the preset position, the CHANGE MOVEMENT POSITION BY OBJECT must be activated. The preset position can be set afterwards by using this object. **Only movement position:** the shutter will move to a certain position (0-100%) when executing the preset (unless otherwise saved in CHANGE MOVEMENT POSITION BY OBJECT); the exact position can be parameterized here. **Only slat position**: not applicable for shutter configuration. Blinds (with slats): the slats will move to a certain position (0-100%), which can be parameterized here. **Movement and slat position**: not applicable for shutter configuration. Blinds (with slats): the blind and the slats will move to a certain position (0-100%), which can be parameterized here.	
Change movement position by object	No function **Only movement position** Only slat position Movement and slat position
No function: this functionality is hidden. **Only movement position**: the absolute position (0-100%) of the shutter can be changed with the "Preset X change move position" object. **Only slat position**: not applicable for shutter configuration. Blinds (with slats): the absolute position (0-100%) of the slats can be changed with the "Preset X change slat position" object. **Movement and slat position**: not applicable for shutter configuration. Blinds (with slats): the absolute position (0-100%) of the blind and the slats can be changed with the "Preset X change move position" and "Preset X change slat position" objects.	
One bit object to save current blind/slat position as the new preset value	**No function** Only movement position Only slat position Movement and slat position
No function: this functionality is hidden. **Only movement position**: This activates a 1 bit object to save only the current movement position as the new preset value by sending a 1 to this object. The slat position will not be saved. **Only slat position**: not applicable for shutter configuration. Blinds (with slats): This activates a 1 bit object to save only the current slat position as the new preset value by sending a 1 to this object. The movement position will not be saved. **Movement and slat position**: not applicable for shutter configuration. Blinds (with slats): This activates a 1 bit object to save the current movement and slat position as the new preset value by sending a 1 to this object.	

µBrick Actuator Series

C) Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Advanced functions / (channel dependent) Alarms

Alarm telegrams are used to block the channel. The reaction of the current channel when any/several of the 8 available alarms have been activated can be configured here:

Parameter	Settings
Alarm 1 … Alarm 8	Nothing **Block channel as is** Move Up Move Down. Move to position Move to preset

Nothing: the channel will not participate in the alarm. Thus, it will not be blocked.
Block channel as is: the channel will be blocked, but not move when activating the alarm. Should the alarm be triggered while the blind is moving, the blind will stop immediately and the current status will be sent to the bus.
Move Up: the channel moves UP. The second relay will be opened; and the first relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position)
Move Down: the channel moves DOWN. The first relay will be opened; and the second relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position). If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN, and thus the remaining time will be calculated accordingly.
Move to position: the shutter will move to a certain position (0-100%) when executing the alarm:

Only movement position: the exact position can be parameterized:

Only slat position: not applicable for shutter configuration.

Blinds (with slats): the exact position of the slats can be parameterized here.

Movement and slat position: not applicable for shutter configuration.

Blinds (with slats): the exact position of the blind and of the slats can be parameterized:

Move to preset: you can select one of the four previously configured PRESETS (Channel/Advanced Functions) to be executed on alarm.

Behaviour at end of all alarms	Nothing Move Up Move Down Move to position Move to preset Set to tracked state

Here you can define the behaviour of the current channel when no alarm is active anymore.

Important note: in the General Settings tab you can configure whether or not the alarms must be acknowledged. The "Behaviour at end of all alarms" will only be executed with no active & acknowledged channel alarms, and if the "disable channel function" is in enabled state. Only then, the channel will be unblocked.

Nothing: the channel will not do anything at the end of all alarms.
Move Up: the channel moves UP. The second relay will be opened; and the first relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position)
Move Down: the channel moves DOWN. The first relay will be opened; and the second relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position). If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN, and thus the remaining time will be calculated accordingly.

Move to position: the shutter will move to a certain position (0-100%) at the end of all alarms.

µBrick Actuator Series

Only movement position: the exact position can be parameterized:

Only slat position: not applicable for shutter configuration.
Blinds (with slats): the exact position of the slats can be parameterized.

Movement and slat position: not applicable for shutter configuration.
Blinds (with slats): the exact position of the blind and of the slats can be parameterized.

Move to preset: you can select one of the four previously configured PRESETS (Channel/Advanced Functions) to be executed at the end of all alarms.
Set to tracked state: while the channel is blocked, the other channel-related objects might receive telegrams. Nevertheless, since the channel is blocked, it does not move.

Even though the actuator does not move, it does register all the absolute position events (not the one bit movements, like up/down, slat up/down) in order to be able to go to the state where it would have been at enabling (if the channel had not been blocked).

Attention! The "Behaviour at the end of all alarms" will only be executed with no active & acknowledged channel alarms, and if the "disable channel function" is in enabled state. Only then, the channel will be unblocked.

D) Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / SLAT PARAMETERS / Advanced functions / Disable

Parameter	Settings
Disable object	**Disable with ON** Disable with OFF
This is the object that can be used to block the channel. The priority of all the disable objects (of all channels together – not individually), when compared with the alarms, can be configured in GENERAL SETTINGS / ALARMS / PRIORITY OF DISABLE OBJECT FOR ALL CHANNELS. **Disable with ON:** the current channel will be blocked with a "1" (ON telegram). **Disable with OFF:** the current channel will be blocked with a "0" (OFF telegram).	
- Reaction on bus voltage recovery	**Enable** Disable Last object status
Attention! Establish the priority in general functions	
Enable: the channel will be enabled. **Disable:** the channel will be blocked. **Last object status:** the status of the Enable object will be saved in the actuator's non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will set the object as it was before the bus failure.	
Behaviour at disabling	**Block channel as is** Move Up Move Down Move to position Move to slat and blind position Move to preset
Block channel as is: the channel will be blocked, but not move on disabling. Should the alarm be triggered while the blind is moving, the blind will stop immediately and the current status will be sent to the bus **Move Up:** the channel moves UP. The second relay will be opened; and the first relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position) **Move Down:** the channel moves DOWN. The first relay will be opened; and the second relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position). If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN, and thus the remaining time will be calculated accordingly. **Move to position:** the shutter will move to a certain position (0-100%) on disabling. The exact position can be parameterized here. **Move to slat and blind position**: not applicable for shutter configuration. Blinds (with slats): the blind and the slats will move to a certain position (0-100%) on disabling. The exact position can be parameterized here. **Move to preset**: you can select one of the four previously configured PRESETS (Channel/Advanced Functions) to be executed on disabling.	
Behaviour at enabling	**Enable and leave channel as is** Move Up Move Down Move to position Move to slat and blind position Move to preset Set to tracked state
Enable and leave channel as is: the channel will not do anything when enabled. **Move Up:** the channel moves UP. The second relay will be opened; and the first relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position) **Move Down:** the channel moves DOWN. The first relay will be opened; and the second relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position). If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN, and thus the remaining time will be calculated accordingly. **Move to position:** the shutter will move to a certain position (0-100%) on enabling. The exact position can be parameterized here.	

Move to slat and blind position: not applicable for shutter configuration.

Blinds (with slats): the blind and the slats will move to a certain position (0-100%) on enabling. The exact position can be parameterized here.

Move to preset: you can select one of the four previously configured PRESETS (Channel/Advanced Functions) to be executed on enabling.
Set to tracked state: while the channel is blocked, the other channel-related objects might receive telegrams. Nevertheless, since the channel is blocked, it does not move.

Even though the actuator does not move, it does register all the absolute position events (not the one bit movements, like up/down, slat up/down) in order to be able to go to the state where it would have been at enabling (if the channel had not been blocked).

Attention! ***Enable channel will trigger the behaviour of the next active (lower priority) alarm. Also the "Behaviour at enabling" will only be executed with no active & acknowledged channel alarms.***

5.2.2 Parameter page: OUTPUTS / Channel X1 (Shutter / blind) / Status shutter / blind

Whenever you choose in OUTPUTS, for channel X "SHUTTER" and then, within the channel, "SHUTTER (WITHOUT SLATS)", the "Status Shutter" tab is automatically activated (and, unlike in the binary outputs, cannot be hidden).

On the other hand, if you choose in "BLIND (WITH SLATS)", the "Status Blind" tab is automatically activated.

In the "Status shutter" and "Status blind" tabs you can define which and when the different status telegrams will be sent.

Parameter	Settings
Send 1 byte position status telegram	**At end of movement** During movement and at end No
At end of movement: only after reaching the commanded position on any movement, will the 1 byte "Status blind position" object send this position. **During movement and at end:** both during the course of the movement and after reaching the commanded position on any movement, the 1 byte "Status blind position" object will send this position. The frequency of sending the status telegram during movement can be adjusted here. **No:** the 1 byte "Status blind position" object will be hidden.	
Send 1 byte slat position status telegram	No **Yes**
When you select "Yes" in this option, the "Status slat position" object will be activated, which can be used to inform about the exact position of the slats after each movement.	
Cyclic sending time for blind/slats position	**No** Yes
If you choose to activate this option, you can adjust the frequency on which: • The 1 byte "Status blind position" (Shutters) object will be sent. • The 1 byte "Status blind position" and the "Status slat position" (Blinds) objects will be sent. Should the slat be set to a new position, this new future position will be sent cyclic and not the current position of the slat during its movement.	
1 bit status object for blind at lower end position	**No** Yes
If you select "Yes" on this menu, the 1 bit "Status blind 100%" object will be activated. Only if the shutter has completed its full (lower-end position) movement (100%), will this object = 1. With any other shutter position, the object value = 0.	
1 bit status object for blind at upper end position	**No** Yes
If you select "Yes" on this menu, the 1 bit "Status blind 0%" object will be activated. Only if the shutter is at its start / upper-end position (0%), will this object = 1. With any other shutter position, the object value = 0.	
Behaviour at disabling	**No** Yes
With this option, the channel's status telegram can also be sent as soon as the device has initialized after bus recovery. You can also configure a delay for sending this status telegram, which can be done in GENERAL SETTINGS / ADVANCED FUNCTIONS / BEHAVIOUR AT BUS RECOVERY / DELAY FOR SENDING ALL STATUS TELEGRAMS.	

6. Parameter page: ADVANCED FUNCTIONS

Tip! REDUCE CONFIG TIME! All repetitive Tab & Sub-Tab parameters (Ex. “Channel A1…X” or “Logic 1…X”…) can be changed at the same time by selecting multiple tabs with “CTRL + Click”.

6.1 Parameter page: Alarms

Parameter	Settings
Alarms	**No** Yes

First of all, in order for the channel-related Alarms to work, the Alarms must be activated by selecting yes.

Then up to 8 alarms to be either “analog” or “digital” can configured

Now, in the Advanced Functions of the channel-dependent alarms which can be found in OUTPUTS/Channel X/Advanced functions/Alarms, you can configure the behaviour of the channel when the alarm objects receive a telegram.

Alarm telegrams are used to block the channel. The reaction of the current channel when any/several of the 8 available alarms have been activated can be configured in the Alarms tab in the output.

Terminology for alarms:
Alarm X enabled / disabled: The alarm can be disabled with the “Alarm X disable” object. This leaves the alarm without any function.

Alarm active / Alarm activated: This means that the alarm has receive a telegram on its “Alarm X” object which triggers the alarm in its active state. This causes the channels (depending on the channel parameters) to be blocked.

Alarm is triggered: if the alarm is activated while it was already active it will not be triggered if “only the first time” is selected in the trigger parameter.

Alarm inactive / Alarm deactivated / Alarm not active / Alarm ended: This means that the alarm has receive a telegram on its “Alarm X” object which ends the alarm in its inactive state.

Channel disabled: Each channel has a “[X] Disable channel” object with which the channel can be blocked.

Channel enabled: Each channel has a “[X] Disable channel” object with which the channel can be enabled. It will only be unblocked though with no active and acknowledged channel alarms

Channel blocked: Due to an active alarm or if the channel was disabled with the “[X] Disable channel” object the channel will be blocked.

Channel unblocked: The channel will only be unblocked with no active and acknowledged channel alarms and if the “disable channel function” is in the enabled state.

Alarm acknowledged: An alarm can only be acknowledged if it is not active. If the acknowledge function is active the channel will have no reaction (no change in the output nor can it be unblocked) until the alarm is acknowledged. This is independent of the “disable channel object” i.e. the alarm can be acknowledged even though the channel is disabled.

µBrick Actuator Series

Example Alarms Table with “Acknowledge needed” active, and “Priority of disable object for all channels” > Alarm 2.

This table describes the different behaviours (on the right of the grey column) with consecutive events (left side of the grey column) The order of the events and their respective behaviours are indicated by a number starting from the first event/behaviour with 1 and counting up with each new event. For example line two:

Event (left side of the grey column)	Behaviour (on the right of the grey column)
1) Alarm 1 is activated	1) Behaviour alarm 1 & Block channel
2) An acknowledge is received	2) No reaction
3) Alarm 1 is deactivated	3) No reaction
4) An acknowledge is received	4) Behaviour at end of all alarms & Unblock Channel

Alarm 1 = 0	Alarm 1 = 1	Disable	Enable	Alarm 2 = 0	Alarm 2 = 1	Ack		Behaviour alarm 1	Behaviour at disable	Behaviour at enable	Behaviour alarm 2	Behaviour at end of all alarms	Block channel	Unblock Channel	No reaction	Alarms ACK but do Nothing
						1									1	
3	1					2, 4		1				4	1	4	2, 3	
2	1					3		1				3	1	3	2	
		1	2						1	2			1	2		
				2	1	3					1	3	1	3	2	
3.1	1	2	4			3.2, 5		1	3.2	4			1	4	2	
3	1	2	4			5		1		4		5	1	5	2, 3, 4	
3.1	1			4	2	3.2, 5		1			3.2	5	1	5	2, 3.1, 4	
3	2	1	5			4		2	1, 4	5			1	5	3	
		2	5	3	1	4			2	5	1		1	5	3	4
		2	4	3	1	5			2		1	5	1	5	3, 4	
6	3	2	5	4	1	7		3	2		1	7	1	7	4, 5, 6	
5	3	2	7	4	1	6		3	2, 6	7	1		1	7	4, 5	6
		2	3	4	1	5			2		1, 3	5	1	5	4	
4.1	3	2	5	6	1	4.2, 7		3	2, 4.2		1, 5	7	1	7	6, 4.1	
3	1	2	5			4		1	4	5			1	5	2, 3	
		2	4	3	1			1	2	4			1		3	

Parameter	Settings
Alarm 1	No **Yes**
By default the first alarm is activated. This option activates or hides the alarm tab with all its parameters.	
Alarm 2…8	**No** Yes
By default the first alarm is deactivated. This option activates or hides the alarm tab with all its parameters.	
Acknowledge needed	Ack. with 0 Ack. with 1 **No**
* Ack. with 0 / 1: ***Attention! Acknowledge will not execute the "Behaviour at end of all alarms" if the "disable channel object" is in disabled state, but if all alarms have ended, they will be acknowledged.***	
By activating this function the alarm must be acknowledged (either with a 1 or with a 0 depending on the above parameter selection) in order to unblock the channel. An alarm can only be acknowledged if it is not active. The channel will have no reaction (no change in the output nor can it be unblocked) until the alarm is acknowledged. This is independent of the "disable channel object" i.e. the alarm can be acknowledged even though the channel is disabled.	
Priority of disable object for all channels	**< Alarm 8** > Alarm 1 > Alarm 2 > Alarm 3 > Alarm 4 > Alarm 5 > Alarm 6 > Alarm 7 > Alarm 8
Each and every channel has a Disable object, which blocks all other functions of the channel. The behaviour at Disabling/Enabling can be configured per channel. The priority of all Disable objects can here be adjusted to have higher/lower priority as the alarms.	

6.1.1 Parameter page: Alarm 1…8

Parameter	Settings
Description	
This enables the integrator to add a personalized description in the text field.	
Type of alarm	**Digital** Analog
Both digital and analog alarms can be used.	

6.2.1 Parameter page: Alarms / Digital

Parameter	Settings
Digital alarm is active when receiving	**On** Off
This parameter is to decide with which useful data of the telegram the alarm will be activated.	
Object to disable Alarm	**No** Yes
The alarm can be disabled with a one bit object. It will be disabled with a 1 and enabled with a 0	
Reaction on bus voltage recovery	**Enable** Disable Last object status
On bus voltage recovery the alarm can be enabled, disabled, or have the same state as before the bus failure depending on the above selection.	
Monitoring time base	**10 s** 1 min 5 min 10 min 1 h
The alarm object must receive a telegram within this time, otherwise the alarm will become active.	
Alarm is triggered	**Always** Only first time
This parameter indicates if the alarm should be triggered each time it is activated or if it should only be triggered the first time. If the alarm is activated while it was already active it will not be triggered if "only the first time" is selected.	

6.2.2 Parameter page: Alarms / Analog

µBrick Actuator Series

Parameter	Settings
Input value Analog alarm	1 byte unsigned 1 byte scaling **2 bytes float** 4 bytes unsigned 4 bytes float
The analog alarms can have any of the above datapoint types. With the analog alarms you only need to have sensors to send the analog values. You are not forced to use the usually very "rigid" logic of a KNX whether station. Apart from not being flexible to create the correct condition one only disposes of the number of threshold of the weather station. On the other hand with this function in the actuator there are much more thresholds.	
Alarm setpoint [x 0.1]	**300**
This is the setpoint of the analog alarm.	
Hysteresis [x 0.1]	**10**
This is the hysteresis of the analog alarm	
Type of Hysteresis (Threshold calculation)	**Setpoint = Upper Threshold** Setpoint = Lower Threshold Setpoint = Symmetric (1/2 between THs)
The hysteresis can be asymmetric or symmetric as can be seen in the above options. If Setpoint = Upper Threshold then the Lower Threshold = Setpoint – Hysteresis If Setpoint = Lower Threshold then the Upper Threshold = Setpoint + Hysteresis If Setpoint = Symmetric (1/2 between THs) then the Upper Threshold = Setpoint + ½ Hysteresis and the Lower Threshold = Setpoint - ½ Hysteresis	
Objects for changing Setpoint/Hysteresis values	**No** Yes

* With Yes

Attention! The end-user parameter values will only be maintained when "Overwrite end-user…" in general tab were set to "Don't overwrite".

Both the setpoint value and the Hysteresis can be changed from the bus. Together with a visualization the customer can adjust each and every threshold to his own criteria. E.g. Wind speed for the awnings, light lux level for the blind position, sun position to move the slats of the blinds, etc.	
Analog alarm is active when	**Exceeding/equal upper threshold** Falling below/equal lower threshold Between upper and lower threshold >/= upper or </= lower threshold
This is to decide when the analog alarm should be active and when it should end (be inactive).	
Object to disable alarm	No **Yes**
The alarm can be disabled with the "Alarm X disable" object. This leaves the alarm without any function.	
Reaction on bus voltage recovery	**Enable** Disable Last object status
On bus voltage recovery the alarm can be enabled, disabled, or have the same state as before the bus failure depending on the above selection.	
Monitoring time base	**10 s** 1 min 5 min 10 min 1 h
The alarm object must receive a telegram within this time, otherwise the alarm will become active.	
Alarm is triggered	Always **Only first time**
This parameter indicates if the alarm should be triggered each time it is activated or if it should only be triggered the first time. If the alarm is activated while it was already active it will not be triggered if "only the first time" is selected.	

6.2 Parameter page: Logics

There are 20 logic functions available

Parameter	Settings
Logics	**No** Yes
The logic functions can be activated here.	

Parameter	Settings
Description	
This enables the integrator to add a personalized description in the text field.	
Type of logic	No function **Boolean** Gate / Filter Mathematical Comparators Converters
One of the above logic functions can be selected.	

6.2.1 Parameter page: Logics / Boolean

Parameter	Settings
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The function can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Type of Boolean function	**AND** NAND OR NOR XOR XNOR
One of the following Boolean logic functions can be configured.	

6.2.1.1 Parameter page: Logics / Boolean / Input

Parameter	Settings
Input 1 Input 2	**Yes** Yes, inverted
The inputs can be activated or inverted	
Input 3 Input 4	**No** Yes Yes, inverted
The inputs can be activated, deactivated or inverted	
Reaction with event on input	**Execute logic** Don't execute logic
The logic can be executed (triggered) with an event on the input or not depending on the above selection. If "Don't execute logic" is selected the input will change and will not execute the logic, but if another input receives a value it will take the received value into account.	
Input constant / value after bus recovery	**Value before bus failure** Read on init after initial delay Set input to 0 Set input to 1
The input can be set to a constant value by the parameter "set input to X" given it is not changed from the bus afterwards It can also read the value from the bus after bus recovery, or be saved on bus failure in order to set this value on bus voltage recovery. When it is set to read the value after bus recovery, and in the output of the logic "Execute on init." is set to "Yes", then the answers of the read requests will not execute the logic. (unless the delay of the read requests is set to be greater than 2 seconds) The output will be sent with the reaction of the "Execute on init." command.	

6.2.1.2 Parameter page: Logics / Boolean / Output

Parameter	Settings

Datapoint type of output	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Sending condition	**On change** Always
In this parameter one can decide when the value must be sent. If the value must change in order to send it or not.	
Send when true	No **Yes**
If a value should be sent when true	
Value when true	1
Set here the value that should be sent when true	
Send when false	No **Yes**
If a value should be sent when false	
Value when false	0
Set here the value that should be sent when false	
Cyclic sending time	**No** Send when true Send when false Both
If a value should be sent cyclically when true, false or both.	
Execute on init	**No** Yes

The function will be executed after bus voltage recovery if “yes” is selected. With “No”: Attention! If No is selected, not even the response of the read on init will execute the logic With “Yes” and the inputs set to read on init, the output is calculated with all response telegrams

6.2.2 Parameter page: Logics / Gate / Filter

Parameter	Settings
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The function can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Reaction on bus voltage recovery of both disable objects	**Enable** Disable Last object status
On bus voltage recovery the logic can be enabled, disabled, or have the same state as before the bus failure depending on the above selection.	

6.2.2.1 Parameter page: Logics / Gate/Filter / Input

Parameter	Settings
Datapoint type	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Reaction of output with event on input	**Always** On change Don’t send telegram
The reaction of output with event on input can be configured with the above options	
Enable / Disable GATE/FILTER	No **En = 1 / Dis = 0**

	En = 0 / Dis = 1
This is the enable / disable input of the gate (not of the logic block) Depending of the above selection the gate will let the values of the input through to the output or not.	
Trigger input to output on en-/disable	**Nothing** Always, on every enable telegram Only when changed from disabled to enabled Always, on every disable telegram Only when changed from enabled to disabled Always, on every en-/disable telegram
The input will be triggered to the output when receiving a telegram on the Enable / disable input independent of the in/out sending conditions. One can decide with this parameter when to do the trigger.	
Input constant / value after bus recovery	**Value before bus failure** Read on init after initial delay Set input to value
The input can be set to a constant value by the parameter “set input to value” given it is not changed from the bus afterwards It can also read the value from the bus after bus recovery, or be saved on bus failure in order to set this value on bus voltage recovery.	

6.2.2.2 Parameter page: Logics / Gate/Filter / Output

Parameter	Settings
Datapoint type of output	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Sending condition	**On change** Always
In this parameter one can decide when the value must be sent. If the value must change in order to send it or not.	
Cyclic sending	**No** Yes
The telegram will be repeated cyclically (with a configurable frequency)	
Output filter	**No** Only let through within range Only let through outside of range
The values to be let through or not (filtered) can be configured here.	
Execute on init	**No** Yes
The function will be executed after bus voltage recovery if “yes” is selected. With “No”: Attention! If No is selected, not even the response of the read on init will execute the logic With “Yes” and the inputs set to read on init, the output is calculated with all response telegrams	

6.2.3 Parameter page: Logics / Mathematical

Parameter	Settings
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The function can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Type of mathematical function	**ADD** SUBSTRACT MULTIPLY DIVIDE MAXIMUM MINIMUM AVERAGE
The type of mathematical function can be selected from one of the options above.	

6.3.3.1 Parameter page: Logics / Mathematical / Input

Parameter	Settings
Input 1 Input 2	No **Yes**
The inputs can be activated or inverted	
Input 3 Input 4	**No** Yes
The inputs can be activated, deactivated or inverted	
Datapoint type of input	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Reaction with event on input	**Execute logic** Don't execute logic
The logic can be executed (triggered) with an event on the input or not depending on the above selection. If "Don't execute logic" is selected the input will change and will not execute the logic, but if another input receives a value it will take the received value into account.	
Input constant / value after bus recovery	**Value before bus failure** Read on init after initial delay Set input to value
The input can be set to a constant value by the parameter "set input to value" given it is not changed from the bus afterwards It can also read the value from the bus after bus recovery, or be saved on bus failure in order to set this value on bus voltage recovery.	

6.2.3.2 Parameter page: Logics / Mathematical / Output

Parameter	Settings
Datapoint type of output	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Sending condition	**On change** Always
In this parameter one can decide when the value must be sent. If the value must change in order to send it or not.	
Cyclic sending	**No** Yes
The telegram will be repeated cyclically (with a configurable frequency)	
Output filter	**No** Only let through within range Only let through outside of range
The values to be let through or not (filtered) can be configured here.	
Execute on init	**No** Yes
The function will be executed after bus voltage recovery if "yes" is selected. With "No": Attention! If No is selected, not even the response of the read on init will execute the logic With "Yes" and the inputs set to read on init, the output is calculated with all response telegrams	

6.2.4 Parameter page: Logics / Comparators

Parameter	Settings
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The function can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	

Type of comparators function	**EQUAL** GREATER SMALLER GREATER OR EQUAL SMALLER OR EQUAL DISTINCT
The type of comparator function can be selected from one of the options above.	

6.2.4.1 Parameter page: Logics / Comparators / Input

Parameter	Settings
Input 1 Input 2	No **Yes**
The inputs can be activated or inverted	
Input 3 Input 4	**No** Yes
The inputs can be activated, deactivated or inverted	
Datapoint type of input	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Reaction with event on input	**Execute logic** Don't execute logic
The logic can be executed (triggered) with an event on the input or not depending on the above selection. If "Don't execute logic" is selected the input will change and will not execute the logic, but if another input receives a value it will take the received value into account.	
Input constant / value after bus recovery	**Value before bus failure** Read on init after initial delay Set input to value
The input can be set to a constant value by the parameter "set input to value" given it is not changed from the bus afterwards It can also read the value from the bus after bus recovery, or be saved on bus failure in order to set this value on bus voltage recovery.	

6.2.4.2 Parameter page: Logics / Comparators / Output

Parameter	Settings
Datapoint type of output	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Sending condition	**On change** Always
In this parameter one can decide when the value must be sent. If the value must change in order to send it or not.	
Send when true	No **Yes**
If a value should be sent when true	
Value when true	1
Set here the value that should be sent when true	
Send when false	No **Yes**
If a value should be sent when false	
Value when false	0
Set here the value that should be sent when false	
Cyclic sending time	**No** Send when true

	Send when false Both
If a value should be sent cyclically when true, false or both.	
Execute on init	**No** Yes
The function will be executed after bus voltage recovery if "yes" is selected. With "No": Attention! If No is selected, not even the response of the read on init will execute the logic With "Yes" and the inputs set to read on init, the output is calculated with all response telegrams	

6.2.5 Parameter page: Logics / Converters

Parameter	Settings
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The function can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	

6.3.5.1 Parameter page: Logics / Converters / Input

Parameter	Settings
Datapoint type of input	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Reaction with event on input	**Execute logic** Don't execute logic
The logic can be executed (triggered) with an event on the input or not depending on the above selection. If "Don't execute logic" is selected the input will change and will not execute the logic, but if another input receives a value it will take the received value into account.	
Input constant / value after bus recovery	**Value before bus failure** Read on init after initial delay Set input to value
The input can be set to a constant value by the parameter "set input to value" given it is not changed from the bus afterwards It can also read the value from the bus after bus recovery, or be saved on bus failure in order to set this value on bus voltage recovery.	

6.2.5.2 Parameter page: Logics / Converters / Output

Parameter	Settings
Datapoint type of output	1 bit 1 byte scaling **1 byte unsigned** 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
For this function one of the above standard KNX datapoint types can be selected.	
Sending condition	**On change** Always
In this parameter one can decide when the value must be sent. If the value must change in order to send it or not.	
Cyclic sending	**No** Yes
The telegram will be repeated cyclically (with a configurable frequency)	
When result value exceeds max. allowed DPT of output value:	Don't send **Send max. value of output** Send value

An overflow is reached when the object value exceeds the maximum value of the selected data point type. For example, the maximum value of a 1 byte unsigned value is 255; therefore, the overflow is reached when the object value exceeds 255. If the result exceeds this maximum DPT value one can select to not send anything, send max. value of output, or send a predefined value.	
When result value is lower than allowed DPT of output value:	Don't send **Send min. value of output** Send absolute value (without sign) Send value
If the result is lower than the minimum value of the DPT one can select to not send anything, send min. value of output, Send absolute value (without sign) or send a predefined value.	
Output filter	**No** Only let through within range Only let through outside of range
The values to be let through or not (filtered) can be configured here.	
Execute on init	**No** Yes
The function will be executed after bus voltage recovery if "yes" is selected. With "No": Attention! If No is selected, not even the response of the read on init will execute the logic With "Yes" and the inputs set to read on init, the output is calculated with all response telegrams	

6.3 Parameter page: Scene controller

Parameter	Settings
Scene controller	**No** Yes
The actuator can also be used as a scene controller with a KNX scene input object (play and record function) and with up to 8 output objects each with its own DPT and values.	

Parameter	Settings
Attention! The end-user parameter values will only be maintained when "Overwrite end-user…" in general tab were set to "Don't overwrite".	
First scene	No **Yes**
Second scene … Tenth scene	**No** Yes
There are 10 scenes which can be individually activated here	

6.2.1 Parameter page: First scene / Tenth scene

Parameter	Settings
Description	
This enables the integrator to add a personalized description in the text field.	
Scene number	**Scene 1** … Scene 64
Each scene can be assigned by this parameter a different input KNX scene number. Any of the 64 possible numbers can be used. The scene number to be received can be configured here. Scene 1 = value 0, Scene 2 = value 1 and so forth up to value Scene 64 = value 63.	
Possible to save scene	No **Yes**
With this selection the scene can be saved. Saving Scene 1 will requires the value 128, Scene 2 requires value 129 and so forth up to Scene 64 requires value 191 to be received in the scene input object.	
Object values are updated with	Read request to bus **Last values stored in the objects**
The values to be used when saving can be configured here, either with a read request to bus or with the last values received in the objects. Thus the user can set the desired values (e.g. using normal pushbuttons or with a visualization) of the loads and then save the new scene with a long press of the button. (according to the KNX scene standard)	
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1

The function can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	
Output value for event 1 … Output value for event 8	**No function** 1 bit 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
Each output can have its own DPT, even 4 byte values.	

6.4 Parameter page: Advanced scene controller

Parameter	Settings
Advanced scene controller	**No** Yes
The actuator can also be used as an advanced scene controller with a free configurable input object (with different DPTs and triggers) and with up to 8 output objects each with its own DPT and values. These outputs can even have a delay between events.	

Parameter	Settings
Attention! The end-user parameter values will only be maintained when "Overwrite end-user…" in general tab were set to "Don't overwrite".	
First scene	No **Yes**
Second scene … Tenth scene	**No** Yes
There are 10 advanced scenes which can be individually activated here	

6.4.1 Parameter page: First scene / Tenth scene

Parameter	Settings
Description	
This enables the integrator to add a personalized description in the text field.	
DPT for Play, Record, Restore and Stop	**1 bit** 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
The input object, unlike the standard KNX scene, can have any of the above DPTs and have different values for the following trigger events: Play, Record, Restore and Stop	
Play value	**0**
Value to start the scene	
Record	**No function** Set record value
Value to record the scene	
Restore	**No function** Set record value
Value to restore the scene. All the previous values of the output objects are always stored in a buffer in order to be able to restore to the previous values before the scene was executed.	
Stop	**No function** Set record value
The scene can have delay between events and can be stopped with this value at any time.	
Enable / Disable object	**No** En = 1 / Dis = 0 En = 0 / Dis = 1
The function can be enabled or disabled by object when selecting this parameter. It can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa.	

Behaviour at reception of new play value while executing scene	**Restart scene** Do nothing
The behaviour at reception of new play value while executing the scene can be configured to either do nothing or to restart the scene.	
Output value for event 1 … Output value for event 8	**No function** 1 bit 1 byte scaling 1 byte unsigned 1 byte signed 2 bytes unsigned 2 bytes signed 2 bytes float 4 bytes unsigned 4 bytes signed 4 bytes float
Each output can have its own DPT, even 4 byte values.	

6.5 Parameter page: Timers

Parameter	Settings
Timers	**No** Yes
The actuator can be used as a timer module with many advanced functions. It can delay any DPT or it can be used as a 1 bit very advanced staircase controller	

Parameter	Settings
Timer 1	No **Yes**
Timer 2 … Timer 10	**No** Yes
There are 10 timers which can be individually activated here.	

6.5.1 Parameter page: Timer 1 / Timer 10

Parameter	Settings
Description	
This enables the integrator to add a personalized description in the text field.	

Timer type	Only “Reaction at OFF” Delay **Staircase** Delay and staircase Only ON (without delay/staircase)
The timer can be used as any of the above timer types. Only the delay can have different DPTs; the rest the of the timer trigger objects are 1 bit objects which will have different behaviours when receiving an ON or OFF respectively. This are the possible actions to be executed when the timer trigger object receives an ON (“1”): Only “Reaction at OFF”: the timer will not be executed. Delay: the channel switches ON after a time delay. Staircase: the channel immediately switches ON and stays ON for the configured staircase time and thereafter switches OFF again. Delay and staircase: the channel switches ON after a time delay and then stays ON for the configured staircase time and thereafter switches OFF again. Only ON (without delay/staircase): the channel immediately switches ON and stays ON.	

6.5.1.1 Parameter page: Timer 1 / 10 / REACTION AT ON

Parameter	Settings
- Staircase time (ON duration) Base	**1 s** 5 s 10 s 1 min 5 min 10 min 1 h
- Staircase time (ON duration) Factor	**60**
Establish here the wished time for the channel to be ON The Staircase time is the period of time during which the actuator channel will be switched ON. After this time elapses, the channel switches OFF again.	
Staircase time Factor changeable by object	**No** Yes

No (default option): staircase time only configurable via parameters.

Yes: this option activates an object to change staircase time factor. As you can see in the picture below, the time Base can be any of the following:

So, if you have selected, for instance, “1 s”, then the values received in this object will be in “seconds”. If you have selected “5 s” though, the values received will be in “seconds” and multiplied by 5 (base “5 s” x value received at object “10” = “50 seconds”). The same rule applies if the Base has been selected in “minutes” or “hours”.

Attention: if you send a 0 to "Timer one change staircase factor" the staircase will switch ON with a “1” and stay ON.

Advanced staircase function	**No** Yes

Here the advanced functions can be activated.

A) Parameter page: Timer 1 / 10 / REACTION AT ON / Advanced staircase function

Parameter	Settings
Multiply staircase	**No** Yes

* With Yes: ***Attention! Total staircase time = staircase time x number of consecutive ON telegrams separated by less than 1 sec. from each other***

Here you can activate the possibility to multiply the staircase time in order to extend the time during which the channel will stay ON. The total staircase ON time is calculated by taking the parameterized staircase time and multiplying it by the number of consecutive ON telegrams received.

Keep in mind that the multiplication telegrams (consecutive ON telegrams) must be separated by less than 1 second from each other. Should more than 1 second elapse between two telegrams, then it will only do the multiplication of the previous pulses received. The telegrams received after this, will be ignored or interpreted as a retrigger timer function (if parameterized), see next parameter “Retrigger timer” option.

This resulting multiplication time will never exceed the maximum staircase time as can be configured in the parameter option “Maximum staircase time Base/Factor”

Practical example: as implied by its name, the staircase time is frequently used in staircases. With the purpose of lowering the costs, instead of using a movement detector for switching ON/OFF, often push buttons are used with the staircase time as defined in the actuator. In order to save energy, the staircase time should be as short as possible, but sometimes you may wish to have the lights longer ON. In this case, this option can be very useful because it allows the end user to easily extend the staircase time by pressing several times (depending on how long the light should stay ON).

Retrigger timer	No **Yes, excluding multiplication** Yes, including multiplication

It is possible to extend the staircase time by retriggering it (in other words, the timer starts counting again from the start) with an ON telegram. But this function will only be executed after more than 1 second has elapsed between the triggering events of the timer (for trigger events less than 1 second, see the behaviour in the section “MULTIPLY STAIRCASE”).

Keep in mind that only the “Staircase time (ON duration)” will be extended. (So if the staircase is configured with an ON delay, when receiving the retrigger telegram it will NOT switch OFF, and the ON delay will be ignored)

If the previous parameter option “Multiply staircase” is activated, the retrigger telegrams will also do the multiplication, given the consecutive ON telegrams are separated by less than 1 second from each other.

No: the staircase will not be retriggered.

Yes, excluding multiplication (default option): this option will retrigger the staircase to be reset to the time (Base/Factor) as configured in the ETS application program.

For example: you have configure the staircase time in the ETS application program to be 1 minute; should the staircase time be, for instance, 1 hour as the result of a previous multiplication (Multiply staircase option), the moment you receive the retrigger telegram it will be reset to 1 minute again.

Yes, including multiplication: this option will retrigger the staircase to be reset to the current staircase time (it could be the parameterized time or the multiplied staircase time).

For example: you have configure the staircase time in the ETS application program to be 1 minute; should the staircase time be, for instance, 1 hour as the result of a previous multiplication (Multiply staircase option), the moment you receive the retrigger telegram it will be reset to 1 hour again.

Warning pulse	**No function** With own output With additional object

The warning pulse is meant to inform the end user about the fact that the staircase time is about to expire.

No function (default option): the light will go OFF without previous warning after the staircase time elapses.

With own output: the same channel will be used for this warning pulse.

According to the default parameters, the output will switch OFF 10 seconds before the end of the staircase time and it will switch ON again 2 seconds thereafter. This creates a short blinking effect as a visual warning.

It is important to be able to configure the OFF time because not all loads can switch OFF immediately (for example, lights using transformers). So, if you have selected 1 second as a warning time, it might not switch OFF at all.

With additional object: this option serves the same purpose of warning before the staircase time elapses. It is specially indicated for those places where the channel can/may not be switched ON and OFF quickly. In these cases, the additional object can send a warning pulse to another channel (different load) just before the end of the staircase time of the main load.

Practical example: let's say this channel is used to control the flood lights of a tennis court via contactor. These lights take long to switch ON again (after they have been switched OFF), which is not energy-efficient nor practical. Therefore, to be able to generate a warning pulse, you can use an additional warning light connected to another channel, which this additional object is linked to.

1 action: ON: the additional object only sends a "1" at the configured point in time before the staircase time elapses.

2 actions : 1st OFF, 2nd ON: the additional object can execute two actions by sending:

- Time before end of staircase for 1st action: a "0" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 2nd action: a "1" at the configured point in time before the staircase time elapses.

2 actions : 1st ON, 2nd OFF: the additional object can execute two actions by sending:

- Time before end of staircase for 1st action: a "1" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 2nd action: a "0" at the configured point in time before the staircase time elapses.

3 actions: 1st OFF, 2nd ON, 3rd OFF (default option): the additional object can execute three actions by sending:

- Time before end of staircase for 1st action: a "0" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 2nd action: a "1" at the configured point in time before the staircase time elapses.
- Time before end of staircase for 3rd action: a "0" at the configured point in time before the staircase time elapses.

6.5.1.2 Parameter page: Timer 1 / 10 / REACTION AT OFF

Parameter	Settings
REACTION AT OFF	No action OFF without delay OFF with delay

Attention! Reaction at OFF cancels the running staircase

This are the possible actions to be executed when the timer trigger object receives an OFF ("0"):

No action: the timer will not be interrupted.

OFF without delay (default option): the channel immediately switches OFF and the timer function is cancelled.

OFF with delay: the channel switches OFF after a time delay. OFF WITH DELAY As soon as the OFF telegram is received, the Timer is cancelled.	
Object to disable timer	Yes, immediately Yes, on ending current timer **No**
The disable object will always react as follows (and cannot be otherwise configured): "1": disable. "0": enable. Yes, immediately: as soon as the Disable object receives a "1", the timer will be cancelled and disabled. This option activates the parameter "Reaction on bus voltage recovery". Yes, on ending current timer: whenever the Disable object receives a "1", the timer will be not cancelled, but disabled. Thus, the current timer will finalize normally. This option activates the parameter "Reaction on bus voltage recovery". No (default option): the disable object, including the "Reaction on bus voltage recovery" will be hidden.	

A) Parameter page: Timer 1 / 10 / REACTION AT OFF / Object to disable timer
With "Object to disable timer:"

- Yes, immediately
- Yes, on ending current timer

Parameter	Settings
Reaction on bus voltage recovery	**Enable** Disable Last object status
On bus voltage recovery the timer can be enabled, disabled, or have the same state as before the bus failure depending on the above selection.	

6.6 Parameter page: Setpoints

Parameter	Settings
Setpoints	**No** Yes
Here the setpoints can be activated. Setpoints can be used as a two-point regulator (2 thresholds) or as an window comparator (2 thresholds + within thresholds)	

6.6.1 Parameter page: Setpoints Tab

Parameter	Settings
Practical example: Thermostat mode control by using 3 setpoints. Setpoint 1 = 22ºC > Enable value = 1 > Comfort mode Setpoint 2 = 20ºC > Enable value = 2 > Standby mode Setpoint 3 = 18ºC > Enable value = 3 > Night mode	
Setpoint 1 … Setpoint 3	No **Yes**
Thermostat controller by using the first 3 setpoints. They have been activated by default and the parameters in each setpoint have been selected individually to build a full KNX room thermostat.	
Setpoint 4 … Setpoint 30	**No** Yes
Here the individual setpoints to use as a Two-point Regulator (2 thresholds), Window comparator (2 thresholds + within thresholds) or simple thermostat can be activated.	

6.6.2 Parameter page: Setpoints 1 … 3

Parameter	Settings
Description	Setpoint 1 default parameter: **Comfort Mode Heat=22ºC, Cool=(22+2)=24ºC** Setpoint 2 default parameter: **Standby Mode Heat=20ºC, Cool=(20+6)=26ºC** Setpoint 3 default parameter: **Night Mode Heat=18ºC, Cool=(18+10)=28ºC**
This enables the integrator to add a personalized description in the text field.	

The actuator does not have a full thermostat module integrated, nevertheless by using 3 setpoints this can be achieved. In order to facilitate the understanding of how to configure the 3 setpoints they have been activated by default and the parameters in each setpoint have been selected individually to build a full KNX room thermostat. It is important to treat these 3 setpoints as “one”. Meaning that the same objects in each of the three setpoints should be linked with the same group address.

E.g. to change the “HVAC mode” i.e. comfort, standby and night mode, the enable object is set to 1 byte and in each setpoint the value to enable the setpoint is different. In the example for Setpoint 1 the enable value is 1, Setpoint 2 the enable value is 2 and Setpoint 3 the enable value is 3. So if the same group address is connected to all three objects, by sending the value 1 the setpoint 1 will be enabled and the other two setpoints disabled. (all other values but the enable value disables the setpoint)

To change the new current setpoint temperature one should, as previously described also connect the same group address to the three “Setpoint X setpoint value/status” objects. Only the enabled setpoint would accept the new setpoint change, thus unlike other room thermostats when changing the current setpoint with the same group address it always changes the value of the current selected mode. Let’s have a detailed look at the default parameter example which uses the first three setpoints:

Thermostat mode control by using 3 setpoints.

1) Setpoint 1 = 22ºC > Enable value = 1 > Heat/Cool = 1 > Mode = Comfort-Heat
2) Setpoint 2 = 20ºC > Enable value = 2 > Heat/Cool = 1 > Mode = Standby-Heat
3) Setpoint 3 = 18ºC > Enable value = 3 > Heat/Cool = 1 > Mode = Night-Heat

4) Setp.1=22ºC+(2ºC Cool offset)=24ºC > Enable=1 > Heat/Cool=0 >Mode=Comfort-Cool
5) Setp.2=20ºC+(6ºC Cool offset)=26ºC > Enable=2 > Heat/Cool=0 >Mode=Standby-Cool
6) Setp.3=18ºC+(10ºC Cool offset)=28ºC > Enable=3 > Heat/Cool=0 >Mode=Night-Cool

As we can see the “Room Thermostat” can be set in 6 states. Now referring to the above states “1) - 6)” let’s see what happens when sending the new setpoint value to all three setpoints at the same time.

Let’s say we start off in state 1) now we send the value 21 as the new setpoint value, this will result in the following:

1) Setpoint 1 = 21ºC > Enable value = 1 > Heat/Cool = 1 > Mode = Comfort-Heat
2) Setpoint 2 = 20ºC > Enable value = 2 > Heat/Cool = 1 > Mode = Standby-Heat
3) Setpoint 3 = 18ºC > Enable value = 3 > Heat/Cool = 1 > Mode = Night-Heat

4) Setp.1=21ºC+(2ºC Cool offset)=23ºC > Enable=1 > Heat/Cool=0 >Mode=Comfort-Cool
5) Setp.2=20ºC+(6ºC Cool offset)=26ºC > Enable=2 > Heat/Cool=0 >Mode=Standby-Cool
6) Setp.3=18ºC+(10ºC Cool offset)=28ºC > Enable=3 > Heat/Cool=0 >Mode=Night-Cool

Now let's say we change to state 2) now we send the value 19 as the new setpoint value, this will result in the following:

1) Setpoint 1 = 21ºC > Enable value = 1 > Heat/Cool = 1 > Mode = Comfort-Heat
2) Setpoint 2 = 19ºC > Enable value = 2 > Heat/Cool = 1 > Mode = Standby-Heat
3) Setpoint 3 = 18ºC > Enable value = 3 > Heat/Cool = 1 > Mode = Night-Heat

4) Setp.1=21ºC+(2ºC Cool offset)=23ºC > Enable=1 > Heat/Cool=0 >Mode=Comfort-Cool
5) Setp.2=19ºC+(6ºC Cool offset)=25ºC > Enable=2 > Heat/Cool=0 >Mode=Standby-Cool
6) Setp.3=18ºC+(10ºC Cool offset)=28ºC > Enable=3 > Heat/Cool=0 >Mode=Night-Cool

Now let's say we change to state 6) now we send the value 27 as the new setpoint value, this will result in the following:

1) Setpoint 1 = 21ºC > Enable value = 1 > Heat/Cool = 1 > Mode = Comfort-Heat
2) Setpoint 2 = 19ºC > Enable value = 2 > Heat/Cool = 1 > Mode = Standby-Heat
3) Setpoint 3 = 17ºC > Enable value = 3 > Heat/Cool = 1 > Mode = Night-Heat

4) Setp.1=21ºC+(2ºC Cool offset)=23ºC > Enable=1 > Heat/Cool=0 >Mode=Comfort-Cool
5) Setp.2=19ºC+(6ºC Cool offset)=25ºC > Enable=2 > Heat/Cool=0 >Mode=Standby-Cool
6) Setp.3=17ºC+(10ºC Cool offset)=27ºC > Enable=3 > Heat/Cool=0 >Mode=Night-Cool

So as can be seen in this last step the setpoint change will always change the current setpoint status (not the parameter value) It does not matter in which KNX HVAC mode or in Heat/Cool state it is in.

This is a big advantage over most KNX room thermostats. To change the setpoint from a visualization you only need one control element to set the desired current setpoint value and it will always correspond to the current setpoint status.

Input value	**By object** Temp. sensor 1 result Temp. sensor 2 result Temp. sensor 3 result Temp. sensor 4 result Temp. sensor 5 result Temp. sensor 6 result
The reference value for the setpoint can be either one of the temperature sensors resulting values (weighted output) of the inputs or it can receive its value from the bus by selecting "By object"	

6.6.2.1 Parameter page: Setpoints 1 … 3 DPT

Parameter	Settings
Datapoint type of setpoint objects	1 byte unsigned 1 byte scaling 2 bytes unsigned **2 bytes float** 4 bytes unsigned 4 bytes float
Attention! The "… setpoint value/status" object can only be changed if the Setpoint is enabled. Initial setpoint status value if Heat/Cool modes are used: Heating = parameter value, Cooling = parameter value + "Cool offset"	
Here the DPT for both the setpoint and the hysteresis can be set. **Setpoint for most of the important DPTs (not only temperature)** This allows for instance in combination with energy meters and visualization systems to set the maximum consumption for each load and use the 4 byte values as a setpoint in order not to exceed the appointed maximum ¼ hour energy values and therefore reduce the monthly costs.	

A) Parameter page: Setpoints 1 … 3 / DPT / X bytes float

Parameter	Settings
Datapoint type of setpoint objects	… **2 bytes float** … 4 bytes float
The usual DPT for temperature values is a 2 byte float value	
Setpoint [x 0.1]	Setpoint 1 default parameter: **220** Setpoint 2 default parameter: **200** Setpoint 3 default parameter: **180**
Here the initial setpoint value can be set. It can also be changed from the bus and depending on the end-user parameters be overwritten or not when downloading with the ETS. **Higher than normal temperature setpoint value;** Use setpoints (as a thermostat) to control high setpoint temperature values. (the most devices in the marked don't allow temp. setpoint higher than 45ºC) Very useful for solar panel installation control.	
Hysteresis [x 0.1]	10

Here the hysteresis value can be set.	
Type of Hysteresis (Threshold calculation)	Setpoint = Upper threshold Setpoint = Lower threshold Setpoint = Symmetric (1/2 between THs) **Heating / Cooling object**
Here the type of hysteresis for the threshold calculation can be selected. When selecting “Setpoint = Upper threshold” the Lower Threshold = Setpoint – Hysteresis (typically for heating) This is typically used for an analogue value that starts off from a lower value and when reaching the higher threshold value sends a telegram to switch the load. E.g. switch off the heating, lower the shades, etc. When selecting “Setpoint = Lower threshold” the Upper Threshold = Setpoint + Hysteresis (typically for cooling) This is typically used for an analogue value that starts off from a higher value and when reaching the lower threshold value sends a telegram to switch the load. E.g. switch off the cooling, switching on a light when getting too dark, etc. When selecting “Setpoint = Symmetric (1/2 between THs)” the Upper Threshold = Setpoint + ½ Hysteresis and the Lower Threshold = Setpoint - ½ Hysteresis. When selecting “Heating / Cooling object” it switches between the first two options by sending to this object a 1 for Heating or a 0 for Cooling. In this case the “reaction exceeding…, …falling…, and …within…” cannot be selected in the parameters. It is fixed to the following: **For Heating:** Reaction exceeding/equal upper threshold = OFF Reaction falling below/equal lower threshold = ON **For Cooling:** Reaction exceeding/equal upper threshold = ON Reaction falling below/equal lower threshold = OFF	
Send output value	**On change** Always
When selecting “On change” the output will only be sent the first time reaching/crossing the threshold. It will only send again when reaching/crossing the other threshold. “Always” on the other hand will send the output on each input event.	
Offset in setpoint for Cooling [x0.1]	Setpoint 1 default parameter: **20** Setpoint 2 default parameter: **60** Setpoint 3 default parameter: **100**
Here the offset of the setpoint temperature when changing to the cool mode can be selected. Example: Assuming the setpoint is 22ºC, when the value in this parameter is 20 (2K), then the setpoint for cooling will be 22 + 2 = 24ºC	
Enable / disable function	**No** Yes
The setpoint can be enabled or disabled by object when selecting this parameter.	
Attention! The end-user parameter values will only be maintained when “Overwrite end-user…” in general tab were set to “Don’t overwrite”.	

A.1) Parameter page: Setpoints 1 … 3 / DPT/ X bytes float / Enable / Disable function

Parameter	Settings
Enable / disable object	1 bit **1 byte unsigned**
The setpoint can be enabled with a 1 bit on/off telegram or with a 1 byte unsigned telegram. The latter can be used for instance to set the HVAC mode.	
Enable / Disable	Setpoint 1 default parameter: **1** Setpoint 2 default parameter: **2** Setpoint 3 default parameter: **3**

When selecting 1 bit, it can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa. When selecting 1 byte to enable the setpoint, the enable value can be set in the parameters. When sending this enable value to the object the setpoint will be enabled, any other value disables the setpoint. When using it for the HVAC mode use one of the following enable values: Comfort mode = 1 Standby mode = 2 Night/saving mode = 3 Frost/Heat protection = 4	
- Reaction on bus voltage recovery	**Enable** Disable Last object status
Whether the setpoint will be active or not on bus voltage recovery can be configured here. On bus voltage recovery the setpoint can be enabled, disabled, or have the same state as before the bus failure depending on the above selection. **Enable:** the setpoint will be enabled. **Disable:** the setpoint will be disabled. **Last object status:** the status of the Enable object will be saved in the actuator's non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will set the object as it was before the bus failure.	
Reaction of output and setpoint at enabling	Nothing Set calculated output Send setpoint **Both**
The reaction of output and setpoint at enabling can be selected to send the Send setpoint, Set calculated output or both the former. This is especially useful to control Air Condition systems as additional heating and/or cooling. Most KNX thermostats don't send the setpoint values with each change (heat/cool, Comfort/Standby/…) to the bus. In order to control a Split unit as an additional cooling via a gateway it is essential to send the new setpoint on each and every change.	
Reaction of output and setpoint at disabling	**Block and send nothing** Block and set output to 0 and send
The reaction of output and setpoint at disabling can be selected to block and send nothing or to block and set output to 0 and send the setpoint value. This is also useful for the above example.	

6.6.3 Parameter page: Setpoints 4 … 30

Parameter	Settings
Description	
This enables the integrator to add a personalized description in the text field.	
Input value	**By object** Temp. sensor 1 result Temp. sensor 2 result Temp. sensor 3 result Temp. sensor 4 result Temp. sensor 5 result Temp. sensor 6 result
The reference value for the setpoint can be either one of the temperature sensors resulting values (weighted output) of the inputs or it can receive its value from the bus by selecting "By object"	

6.6.3.1 Parameter page: Setpoints 4 … 30 DPT

Parameter	Settings
Datapoint type of setpoint objects	1 byte unsigned 1 byte scaling 2 bytes unsigned **2 bytes float** 4 bytes unsigned 4 bytes float
Attention! The "… setpoint value/status" object can only be changed if the Setpoint is enabled. Initial setpoint status value if Heat/Cool modes are used: Heating = parameter value, Cooling = parameter value + "Cool offset"	

Here the DPT for both the setpoint and the hysteresis can be set.

Setpoint for most of the important DPTs (not only temperature) This allows for instance in combination with energy meters and visualization systems to set the maximum consumption for each load and use the 4 byte values as a setpoint in order not to exceed the appointed maximum ¼ hour energy values and therefore reduce the monthly costs.

A) Parameter page: Setpoints 1 … 3 / DPT / X bytes float

Parameter	Settings
Datapoint type of setpoint objects	… **2 bytes float** … 4 bytes float
Setpoint [x 0.1]	220
Here the initial setpoint value can be set. It can also be changed from the bus and depending on the end-user parameters be overwritten or not when downloading with the ETS. **Higher than normal temperature setpoint value;** Using setpoints (as a thermostat) to control high setpoints temperature values (the most devices in the marked don't allow temp. setpoint higher than 45ºC. Very useful for solar panel installation control.	
Hysteresis [x 0.1]	10
Here the hysteresis value can be set.	
Type of Hysteresis (Threshold calculation)	**Setpoint = Upper threshold** Setpoint = Lower threshold Setpoint = Symmetric (1/2 between THs) Heating / Cooling object
Here the type of hysteresis for the threshold calculation can be selected. When selecting "Setpoint = Upper threshold" the Lower Threshold = Setpoint – Hysteresis (typically for heating) This is typically used for an analogue value that starts off from a lower value and when reaching the higher threshold value sends a telegram to switch the load. E.g. switch off the heating, lower the shades, etc. When selecting "Setpoint = Lower threshold" the Upper Threshold = Setpoint + Hysteresis (typically for cooling) This is typically used for an analogue value that starts off from a higher value and when reaching the lower threshold value sends a telegram to switch the load. E.g. switch off the cooling, switching on a light when getting too dark, etc. When selecting "Setpoint = Symmetric (1/2 between THs)" the Upper Threshold = Setpoint + ½ Hysteresis and the Lower Threshold = Setpoint - ½ Hysteresis. When selecting "Heating / Cooling object" it switches between the first two options by sending to this object a 1 for Heating or a 0 for Cooling. In this case the "reaction exceeding…, …falling…, and …within…" cannot be selected in the parameters. It is fixed to the following: **For Heating:** Reaction exceeding/equal upper threshold = OFF Reaction falling below/equal lower threshold = ON **For Cooling:** Reaction exceeding/equal upper threshold = ON Reaction falling below/equal lower threshold = OFF	
Reaction exceeding/equal upper threshold	No reaction On **Off** On, first time exceeding Off, first time exceeding
Here the reaction exceeding/equal upper threshold can be set.	
Reaction falling below/equal lower threshold	No reaction **On** Off On, first time falling below Off, first time falling below
Here the reaction falling below/equal lower threshold can be set.	
Reaction within threshold	**No reaction** On Off On, first time entering Off, first time entering

Here the reaction within threshold can be set	
Enable / disable function	**No** Yes
The setpoint can be enabled or disabled by object when selecting this parameter.	
Attention! The end-user parameter values will only be maintained when "Overwrite end-user…" in general tab were set to "Don't overwrite".	

A.1) Parameter page: Setpoints 1 … 3 / DPT/ X bytes float / Enable / Disable function

Parameter	Settings
Enable / disable object	**1 bit** 1 byte unsigned
The setpoint can be enabled with a 1 bit on/off telegram or with a 1 byte unsigned telegram. The latter can be used for instance to set the HVAC mode.	
Enable / Disable	**En =1 / Dis = 0** En =0 / Dis = 1
When selecting 1 bit, it can be configured to enable with an ON telegram and to disable with an OFF telegram or vice versa. When selecting 1 byte to enable the setpoint, the enable value can be set in the parameters. When sending this enable value to the object the setpoint will be enabled, any other value disables the setpoint. When using it for the HVAC mode use one of the following enable values: Comfort mode = 1 Standby mode = 2 Night/saving mode = 3 Frost/Heat protection = 4	
- Reaction on bus voltage recovery	**Enable** Disable Last object status
Whether the setpoint will be active or not on bus voltage recovery can be configured here. On bus voltage recovery the setpoint can be enabled, disabled, or have the same state as before the bus failure depending on the above selection. **Enable:** the setpoint will be enabled. **Disable:** the setpoint will be disabled. **Last object status:** the status of the Enable object will be saved in the actuator's non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will set the object as it was before the bus failure.	
Reaction of output and setpoint at enabling	**Nothing** Set calculated output Send setpoint Both
The reaction of output and setpoint at enabling can be selected to send the Send setpoint, Set calculated output or both the former. This is especially useful to control Air Condition systems as additional heating and/or cooling. Most KNX thermostats don't send the setpoint values with each change (heat/cool, Comfort/Standby/…) to the bus. In order to control a Split unit as an additional cooling via a gateway it is essential to send the new setpoint on each and every change.	
Reaction of output and setpoint at disabling	**Block and send nothing** Block and set output to 0 and send
The reaction of output and setpoint at disabling can be selected to block and send nothing or to block and set output to 0 and send the setpoint value. This is also useful for the above example.	

6.7 Parameter page: Internal variables

Parameter	Settings
Internal variables	**No** Yes

This can be used to make internal links like the links done by using group addresses but with the main difference that they are not sent to the bus.

Only output objects can be linked to input objects. Care should be taken to link only objects with the same DPT, this must be checked by the integrator, it is not checked by the application program. Should they have different sizes it will not work.

Parameter	Settings
Internal variables 1…10	No **Yes**
Internal variables 11…20 Internal variables 21…30 Internal variables 31…40 Internal variables 41…50	**No** Yes
Attention! It is recommended to only use variables for internal links. If group addresses are also linked, execution will take longer.	
A total of 50 internal links can be done	

6.7.1 Parameter page: Variables 1…10

Parameter	Settings
Description	
This enables the integrator to add a personalized description in the text field.	

Parameter	Settings
Variable 1	No **Yes**
Variable 2 … Variable 10	**No** Yes
There are a total of 10 variable per page	

6.7.2 Parameter page: Variables 1…10 / Output object

Parameter	Settings
Output object to send variable	General **Switching channels** Blind channels Logic Scenes Advanced scenes Timers Setpoints
In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	

Parameter	Settings
Output object to send variable	General
In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	
Object name	**Central cyclic telegram for monitoring** Telegram at bus recovery
In order to find and select the output object to be linked with the input object one has different filters. This is the first subfilter where all the sub functions of the previously selected main function of the actuator are listed.	

Parameter	Settings
Output object to send variable	**Switching channels**
In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	
Select channel	**A1** A2 B1 B2 C1 C2
In order to find and select the output object to be linked with the input object one has different filters. This is the first sub-	

filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Switching status** RunHour counter RunHour counter alarm RunHour counter value at reset Switching counter Switching counter alarm Switching counter value at reset Timer 1 warning pulse Timer 2 warning pulse
In order to find and select the output object to be linked with the input object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Output object to send variable	Blind channels
In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	
Select channel	**A** B C
In order to find and select the output object to be linked with the input object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Status blind position** Status blind 100% Status blind 0% Status slat position
In order to find and select the output object to be linked with the input object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Output object to send variable	Logics
In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	
Select logic	**Logic 1** … Logic 20
In order to find and select the output object to be linked with the input object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Logic output**
In order to find and select the output object to be linked with the input object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Output object to send variable	Scenes
In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	
Select KNX scene	**Scene 1** … Scene 10
In order to find and select the output object to be linked with the input object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Scene event 1** … Scene event 8
In order to find and select the output object to be linked with the input object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Output object to send variable	Advanced scenes

In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	
Select flexible scene	**Scene 1** … Scene 10
In order to find and select the output object to be linked with the input object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Advanced scene event 1** … Advanced scene event 8
In order to find and select the output object to be linked with the input object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Output object to send variable	Timers
In order to find and select the output object to be linked with the input object one has different filters. This is the main filter where all main functions of the actuator are listed. (except for the inputs – they cannot be linked with internal variables)	
Select timer	**Timer 1** … Timer 10
In order to find and select the output object to be linked with the input object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Timer warning pulse** Timer output
In order to find and select the output object to be linked with the input object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Output object to send variable	Setpoints

Select setpoint	**Setpoint 1** … Setpoint 30
In order to find and select the output object to be linked with the input object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Setpoint output regulator**
In order to find and select the output object to be linked with the input object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

6.7.3 Parameter page: Variables 1…10 / Input object

Parameter	Settings
Input object to send variable	General **Switching channels** Blind channels Alarms Logic Scenes Advanced scenes Timers Setpoints
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	

Parameter	Settings
Input object to send variable	General
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Object name	**Central switching/move blind** Central move Manual control disable

In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.

Parameter	Settings
Input object to send variable	**Switching channels**
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select channel	**A1** A2 B1 B2 C1 C2
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Switching** Switching toggle / inverted RunHour counter threshold RunHour counter reset Switching counter threshold Switching counter reset Scene number Scene disable Timer 1 trigger Timer 1 change staircase factor Timer 1 disable Timer 2 trigger Timer 2 change staircase factor Timer 2 disable Disable channel
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Input object to send variable	Blind channels
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select channel	**A** B C
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Move** Stop (Blind = Stop/Step) Move to position Move to slat Change upper limit Change lower limit Preset 1 execute Preset 2 execute Preset 3 execute Preset 4 execute Preset 1 change move position Preset 2 change move position Preset 3 change move position Preset 4 change move position Preset 1 change slat position Preset 2 change slat position Preset 3 change slat position Preset 4 change slat position Preset 1 save Preset 2 save Preset 3 save Preset 4 save Scene number Scene disable Disable function Move inverted
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

µBrick Actuator Series

Parameter	Settings
Input object to send variable	Alarms
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select alarm	**Alarm 1** … Alarm 8
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Alarm** Alarm setpoint Alarm hysteresis Alarm disable
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Input object to send variable	Logics
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select logic	**Logic 1** … Logic 20
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Logic disable** Logic input 1 Logic input 2 / Enable Gate Logic input 3 Logic input 4
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Input object to send variable	Scenes
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select KNX scene	**Scene 1** … Scene 10
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Scene input** Scene disable
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Input object to send variable	Advanced scenes
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select flexible scene	**Scene 1** … Scene 10
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Advanced scene input** Advanced scene disable
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

µBrick Actuator Series

Parameter	Settings
Input object to send variable	Timers
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select timer	**Timer 1** … Timer 10
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Timer trigger** Timer change staircase factor Timer disable
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

Parameter	Settings
Input object to send variable	Setpoints
In order to find and select the input object to be linked with the output object one has different filters. This is the main filter where all main functions of the actuator are listed. (Except for the inputs – they cannot be linked with internal variables)	
Select setpoint	**Setpoint 1** … Setpoint 30
In order to find and select the input object to be linked with the output object one has different filters. This is the first sub-filter where all the sub functions of the previously selected main function of the actuator are listed.	
Object name	**Setpoint Heat / Cool** Setpoint disable Setpoint value/status Setpoint input ext. sensor value
In order to find and select the input object to be linked with the output object one has different filters. This is the second sub-filter where all the secondary sub functions of the previously selected sub-function of the actuator are listed.	

6.8 Parameter page: Overwrite end-user parameter values at download

Parameter	Settings
Overwrite end-user parameter values at download	No **Yes** Custom
It is very important for the end user to be able to change (via dedicated objects linked, for instance, to a visualization) certain settings of his/her KNX installation. This actuator allows for these changes to be maintained even when downloading the application program with the ETS again. If no end-user parameters should be downloaded the "No" option should be selected. But it is also possible by selecting "Custom" to individually decide whether or not the end-user parameters should be downloaded.	

6.8.1 Parameter page: ENDUSER PARAMETERS

Parameter	Settings
Attention! For blind selection only Channel_1 parameters are used. In this case ignore parameters for Channel_2!	
The channels always are either two binary channels or one shutter/blind channel. It is done like this to reduce the needed parameters.	

6.8.1.1 Parameter page: ENDUSER PARAMETERS / ADVANCED FUNCTIONS

A) Parameter page: ADVANCED FUNCTIONS / Alarms

Parameter	Settings
Alarms	**Overwrite complete module** Overwrite individually Don't overwrite
If none of the Alarm end-user parameters should be downloaded the "Don't overwrite" option should be selected. But it is also possible by selecting "Overwrite individually" to individually decide whether or not the end-user parameters of any one of the 8 Alarms should be downloaded.	

A.1) Parameter page: ADVANCED FUNCTIONS / Alarms / Overwrite individually

Parameter	Settings
Alarms	Overwrite individually
- Alarm 1 … - Alarm 8	Overwrite Don't overwrite
Select here whether to overwrite or not	

B) Parameter page: ADVANCED FUNCTIONS / Scenes

Parameter	Settings
Scenes	**Overwrite complete module** Overwrite individually Don't overwrite
If none of the Scene end-user parameters should be downloaded the "Don't overwrite" option should be selected. But it is also possible by selecting "Overwrite individually" to individually decide whether or not the end-user parameters of any one of the 10 scenes should be downloaded.	

B.1) Parameter page: ADVANCED FUNCTIONS / Scenes / Overwrite individually

Parameter	Settings
Scenes	Overwrite individually
- First scene … - Tenth scene	Overwrite Don't overwrite
Select here whether to overwrite or not	

C) Parameter page: ADVANCED FUNCTIONS / Advanced scenes

Parameter	Settings
Advanced scenes	**Overwrite complete module** Overwrite individually Don't overwrite
If none of the Advanced Scene end-user parameters should be downloaded the "Don't overwrite" option should be selected. But it is also possible by selecting "Overwrite individually" to individually decide whether or not the end-user parameters of any one of the 10 Advanced scenes should be downloaded.	

C.1) Parameter page: ADVANCED FUNCTIONS / Advanced scenes / Overwrite individually

Parameter	Settings
Alarms	Overwrite individually
- First scene … - Tenth scene	Overwrite Don't overwrite
Select here whether to overwrite or not	

D) Parameter page: ADVANCED FUNCTIONS / Timers

Parameter	Settings
Timers	**Overwrite complete module** Overwrite individually Don't overwrite
If none of the Timers end-user parameters should be downloaded the "Don't overwrite" option should be selected. But it is also possible by selecting "Overwrite individually" to individually decide whether or not the end-user parameters of any one of the 10 Timers should be downloaded.	

D.1) Parameter page: ADVANCED FUNCTIONS / Advanced scenes / Overwrite individually

Parameter	Settings
Timers	Overwrite individually
- Timer 1 … - Timer 10	Overwrite Don't overwrite
Select here whether to overwrite or not	

E) Parameter page: ADVANCED FUNCTIONS / Setpoints

Parameter	Settings
Setpoints	**Overwrite complete module** Overwrite individually Don't overwrite
If none of the Setpoints end-user parameters should be downloaded the "Don't overwrite" option should be selected. But it is also possible by selecting "Overwrite individually" to individually decide whether or not the end-user parameters of any one of the 30 Setpoints should be downloaded.	

E.1) Parameter page: ADVANCED FUNCTIONS / Setpoints / Overwrite individually

Parameter	Settings
Setpoints	Overwrite individually
- Setpoint 1 … - Setpoint 30	Overwrite Don't overwrite
Select here whether to overwrite or not	

6.8.1.2 Parameter page: ENDUSER PARAMETERS / OUTPUTS

Parameter	Settings
OUTPUTS	**Overwrite all channels** Overwrite individually Don't overwrite
If none of the binary and blind outputs end-user parameters should be downloaded the "Don't overwrite" option should be selected. But it is also possible by selecting "Overwrite individually" to individually decide whether or not the end-user parameters of any one of the binary and blind outputs parameters should be downloaded.	

A) Parameter page: ENDUSER PARAMETERS / OUTPUTS / CHANNEL A1… C1 (BINNARY / CHANNEL A BLIND)

Parameter	Settings
OUTPUTS	Overwrite individually
- Scenes	Overwrite Don't overwrite
Select here whether to overwrite or not	
- Counters	Overwrite Don't overwrite
Select here whether to overwrite or not	
- Presets / Limits (only for shutter/blind)	Overwrite Don't overwrite
Select here whether to overwrite or not	

µBrick Actuator Series

B) Parameter page: ENDUSER PARAMETERS / OUTPUTS / CHANNEL A2… C2 (ONLY BINARY)

Parameter	Settings
OUTPUTS	Overwrite individually
- Scenes	Overwrite Don't overwrite
Select here whether to overwrite or not	
- Counters	Overwrite Don't overwrite
Select here whether to overwrite or not	

6.9 Parameter page: Central sending object for monitoring device

Parameter	Settings
Central sending object for monitoring device	**No** Yes
This activates a central cyclic sending object which can be used to monitor if the device is still sending this telegram. This way a KNX line and or the actuator can be supervised if they are still reachable.	

Parameter	Settings
- Sending period (0=only answer) min.	**0**
The cyclic sending rate can be introduced here, should the object be polled it is not necessary to send it cyclically and therefore it can be set to zero. Then this object will only answer to read requests.	

6.10 Parameter page: Behaviour at bus recovery

Parameter	Settings
Behaviour at bus recovery	**No** Yes
The behaviour at bus voltage failure and recovery can be established in most parts (outputs, inputs, advanced functions) in the application program of the actuator, but the sending delays and frequencies can be adjusted here.	

Parameter	Settings
- Send telegram for external use	**No** Yes
It is very usual to have to do different actions when the KNX devices are powered up, like a scene to establish some default parameters (establish temperature setpoint values, trigger a scene, reset a variable, etc...). By activating this function the actuator will send a telegram with a fixed value to the bus after bus recovery. The DPT can also be selected to be: 1 bit, 1 byte unsigned, 1 byte scaling and 2 byte float.	
- Delay for sending all status telegrams	Immediately 1 s **5 s** 10 s 20 s 30 s 1 min 3 min 5 min 10 min
The behaviour at bus voltage failure and recovery can be established in most parts (outputs, inputs, advanced functions) in the application program of the actuator, which could cause generating status telegrams after recovery of the bus voltage, but some devices might take longer to start-up (like touch displays, visualization servers, etc.). In these cases the delay for sending the status telegrams can be set here.	
- Delay for all initial read request and execute on init commands	Immediately 1 s 5 s **10 s** 20 s 30 s 1 min 3 min 5 min 10 min
The delay for all initial read request and execute on initialization commands can be set here.	
- Delay between read request / status telegrams	Immediately **500 ms** 1 s 2 s
Should the behaviour on bus voltage return be configured in many places in the actuator, this could cause multiple telegrams to the bus be sent at the same time. For this not to happen one can select here the delay between telegrams sent to the bus after bus recovery.	

µBrick Actuator Series

7. Firmware version and update

If there is a new firmware available, it can be updated via a micro SD card in only a couple of seconds.

Procedure:

1) Remove the bus connector of the device leaving it without bus voltage.

2) Copy the xxxxx.bin (e.g. for the uBrick io66 device the file would be: 1_io66.bin) file to the micro SD card and put it into the micro SD card slot of the device.

3) Press the ETS physical address programing button next to the bus connector of the device

4) Without releasing the button plug in the bus connection while maintaining to hold the button until the programming LED starts to flash and then release it (before it stops to flash)

5) Finished! Now the ETS application program can be download by using the normal procedure using the ETS.

Attention! Never insert the micro SD Card when the device is connected to the KNX bus voltage! This could cause the device to reset without storing the variables previously to the Flash memory. Thus all these variables (e.g. counter values, scene values …) will be lost.

8. Reset to conditions at delivery

To reset the device to its original settings, repeat the same procedure as above using the last valid firmware.

This leads to a factory reset. All device settings return to their status at delivery and the device has the physical address 15.15.255.

µBrick Actuator Series

9. ANNEXES

ANNEX 1: MANUAL CONTROL

The µBrick actuator has 2 push buttons and status LEDs on the front side:

- These buttons can be activated to control each and every channel individually if you select “yes” in the relevant parameter options in Binary outputs and/or Shutter/Blinds.
- The LEDs are arranged in two rows, whereas the LEDs represent:
  - o For Binary outputs:
    - The top row: channels A1, B1, C1, etc.
    - The bottom row: channels A2, B2, C2, etc.
  - o For Shutter/blinds:
    - The top row: channel’s first relay A-UP, B-UP, C-UP, etc.
    - The bottom row: channel’s second relay A-DOWN, B- DOWN, C- DOWN, etc.

PARAMETER MODE

<table>
<tr><td colspan="4">MANUAL CONTROL – PARAMETER MODE

The Parameter Mode allows you to control all the channels of the actuator as configured in the ETS.
The Action simulates a telegram received at the switching object of the selected channel.</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">SHORT PRESS – Channel Selection

  Short LED blinking</td><td colspan="2">LONG PRESS – Action

  LED blinks once off to confirm the action.</td></tr>
<tr><td>BINARY</td><td>SHUTTER/BLIND</td></tr>
<tr><td>TOP push button
←</td><td>- First press action: only indicates current channel
- Consecutive pressing actions (< 0,666 sec. apart): Moves to the left</td><td>Sends OFF command “0” to the “Switching” object

  LED = OFF (indicates channel status)</td><td>- First press action: Sends a DOWN command “1” to the “Move” object.
- Next press action (while shutter/blind is moving) of same button: sends a Stop command to the “Stop…” object.

  LED blinks while moving DOWN during parameterized time</td></tr>
<tr><td>BOTTOM push button
→</td><td>- First press action: only indicates current channel
- Consecutive pressing actions (< 0,666 sec. apart): Moves to the right</td><td>Sends ON command “1” to the “Switching” object

  LED = ON (indicates channel status)</td><td>- First press action: Sends an UP command “0” to the “Move” object.
- Next press action (while shutter/blind is moving) of same button: sends a Stop command to the “Stop…” object.

-  LED blinks while moving UP during parameterized time</td></tr>
</table>

µBrick Actuator Series

TEST MODE

MANUAL CONTROL – TEST MODE

The Test Mode allows you to test all the loads/wiring connected to the channels. It is independent from the ETS configuration of the actuator (since the "Manual Control / Param mode + Test mode" is a default option, you can use the Test mode even before programming the actuator).

Important note: should a blind/shutter be connected to a channel, the 2 channels may never be closed at the same time. Therefore, even in Test mode, if the channel is configured as a blind, this safety measure is implemented (See XX). For this reason, it is better to first commission the OUTPUT: CHANNEL TYPE SELECTION before using the Test mode.

To change into the test mode, both of the buttons must be pressed for 2 seconds. To change back to the normal "Parameter Mode" the same procedure should be repeated. Be aware by changing back to "Parameter Mode" the device will restart. Also after the device has restarted and if the channel is configured to be a blind channel, it will do a calibration movement on the first movement command.

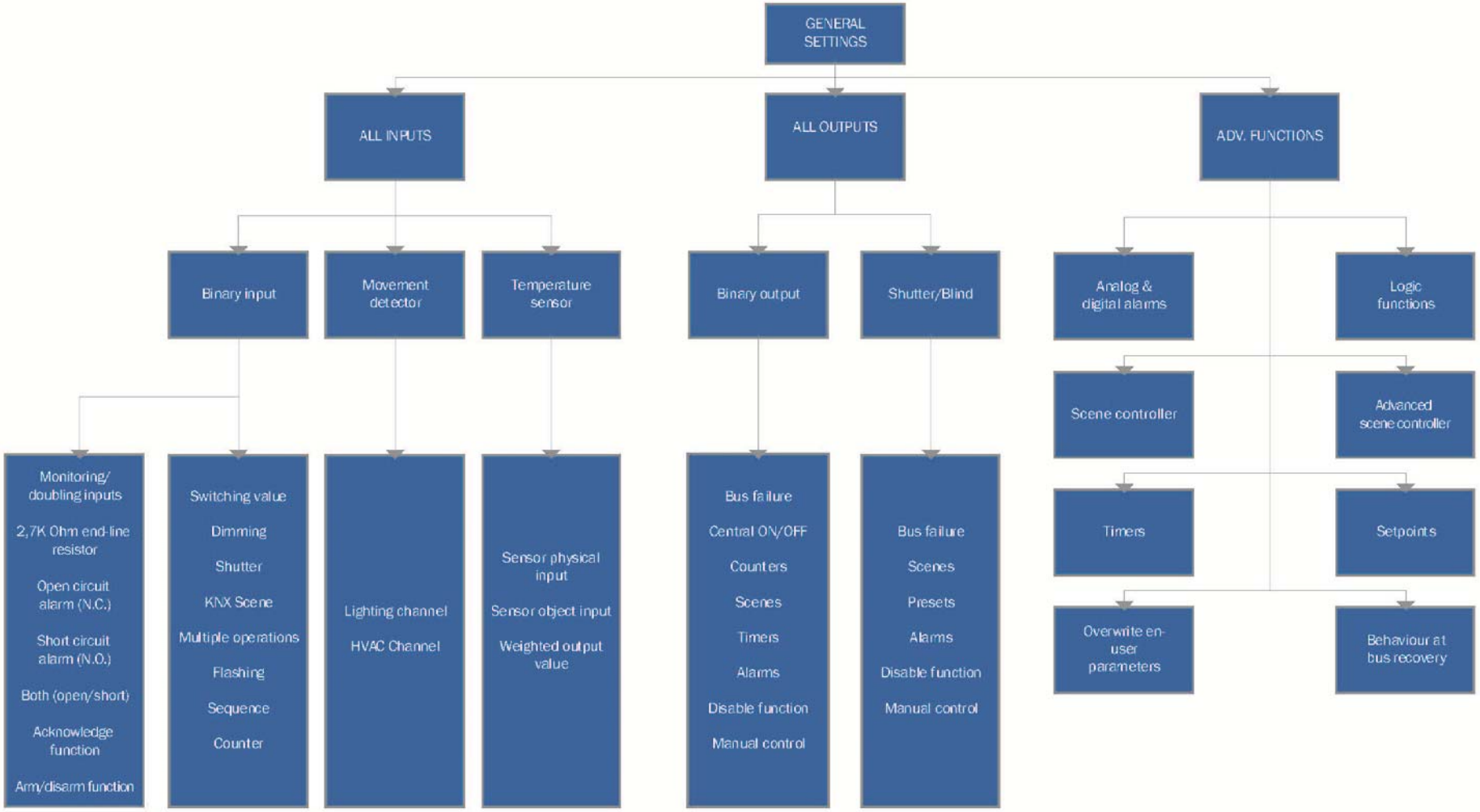
 In order to indicate that the actuator is in Manual Control / Test Mode, the LED of the selected channel is continuously making a short blinking action every second; no matter whether the channel is ON (LED ON) or OFF (LED OFF).

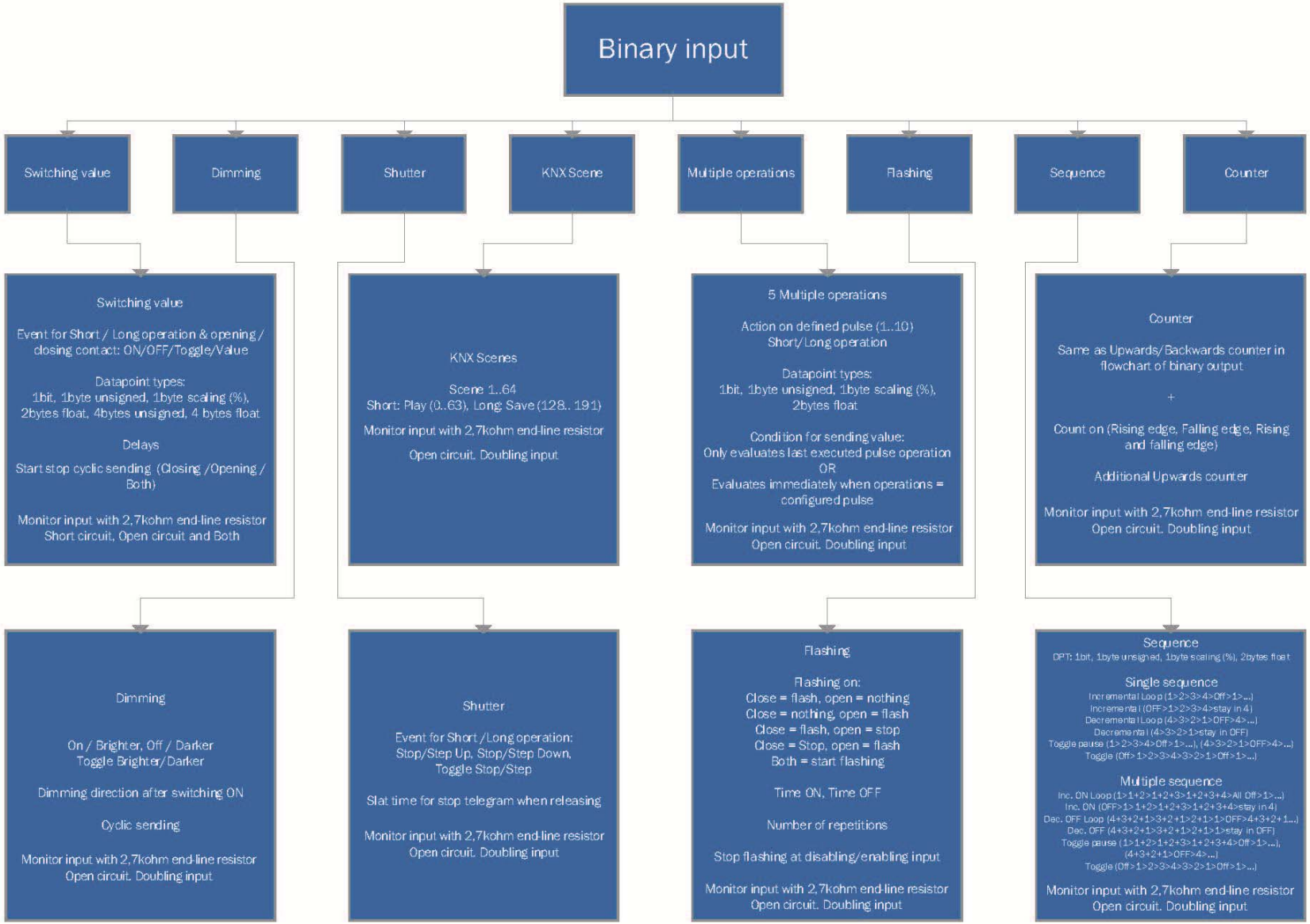
The Action switches/moves the channel, as you can see in the table below:

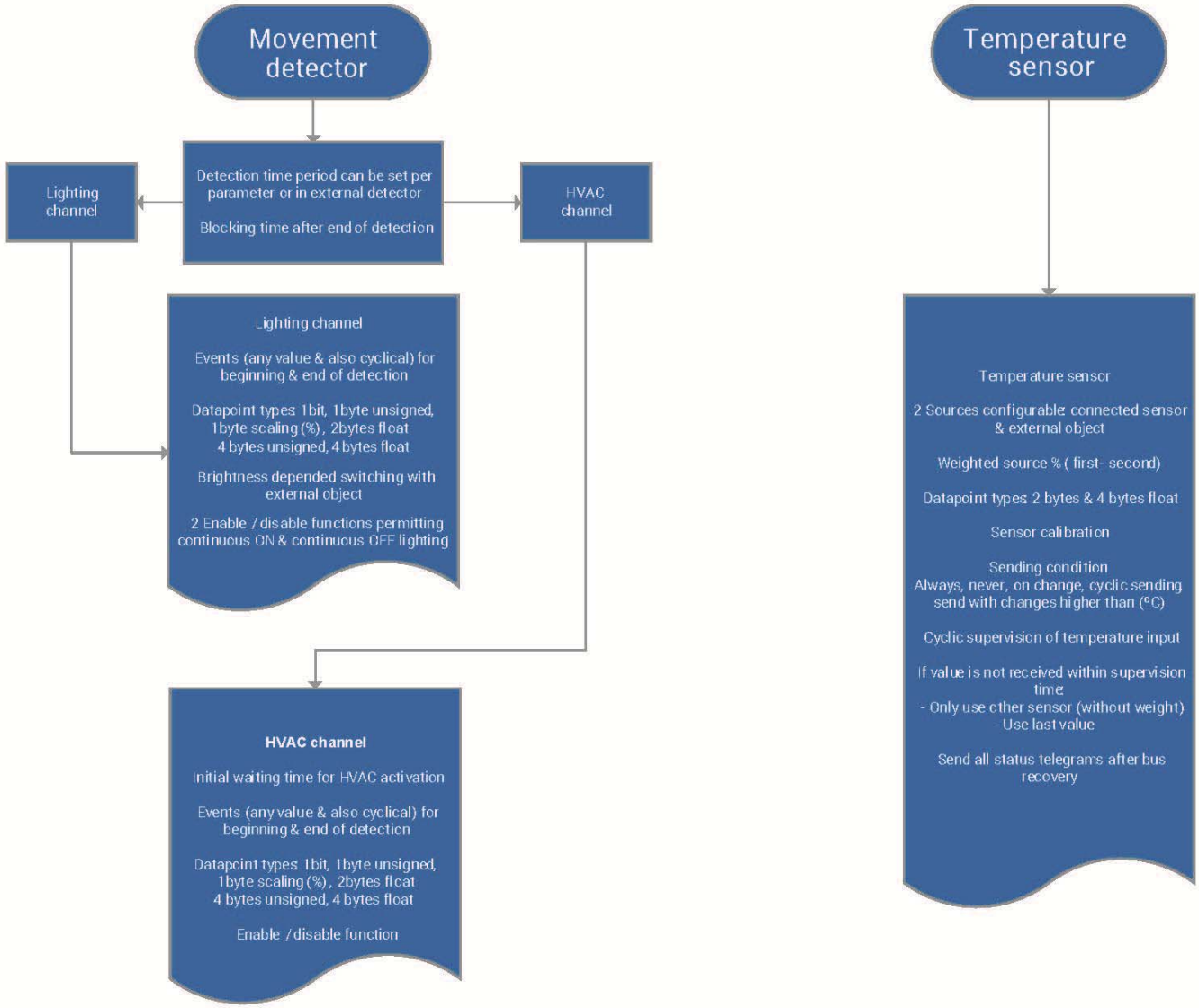
	SHORT PRESS – Channel Selection  Short LED blinking	**LONG PRESS – Action**  LED blinks once off to confirm the action.	
		BINARY	**SHUTTER/BLIND**
TOP push button ←	- First pressing: only indicates current channel - Consecutive pressing actions (not more than 0,666 sec. apart): Moves to the left	Switches OFF  LED = OFF (indicates channel status)	- Long press: Moves DOWN - Release: Stops  LED blinks while moving DOWN
BOTTOM push button →	- First pressing: only indicates current channel - Consecutive pressing actions (not more than 0,666 sec. apart): Moves to the right	Switches ON  LED = ON (indicates channel status)	- Long press: Moves UP - Release: Stops -  LED blinks while moving UP

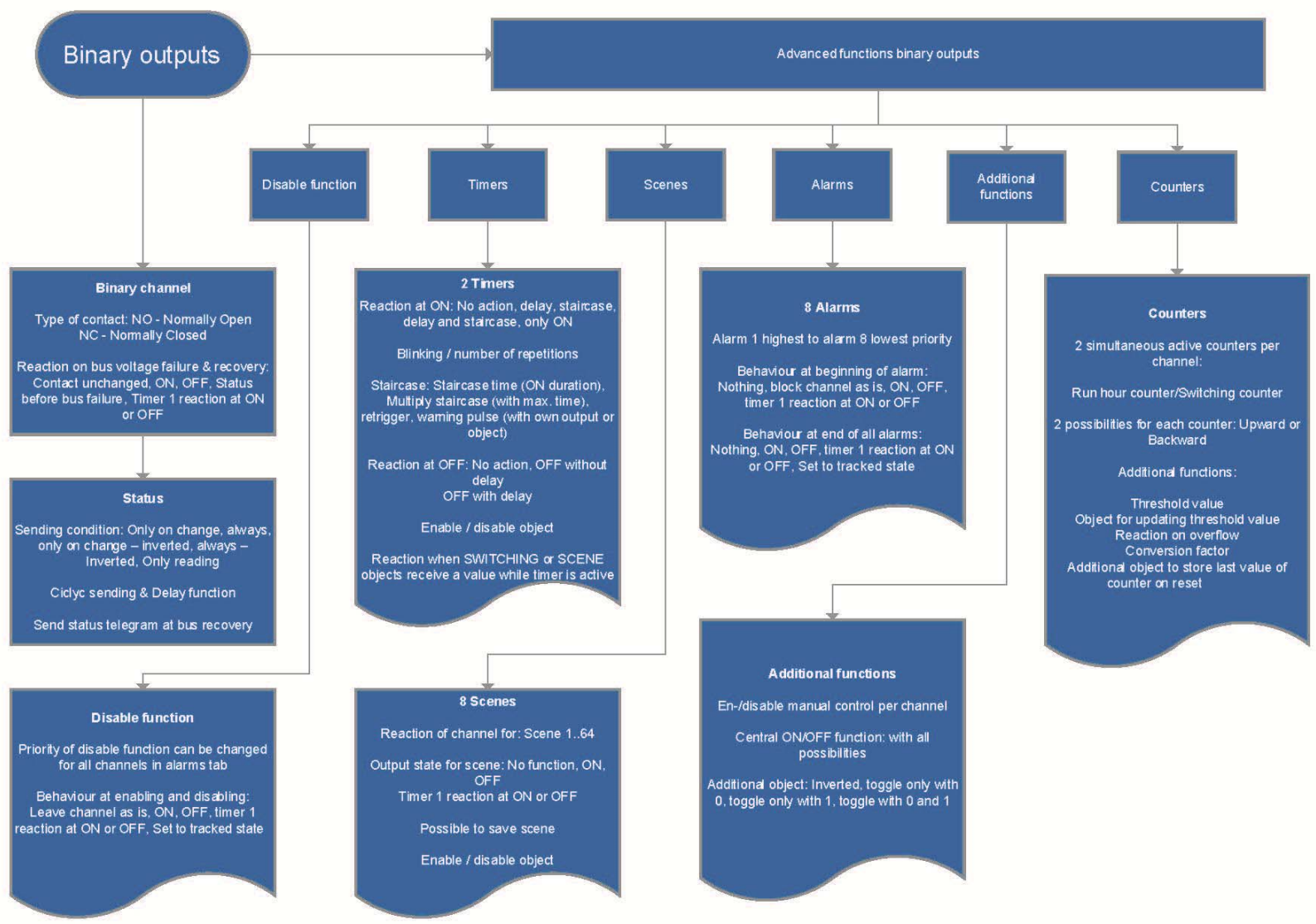
µBrick Actuator Series

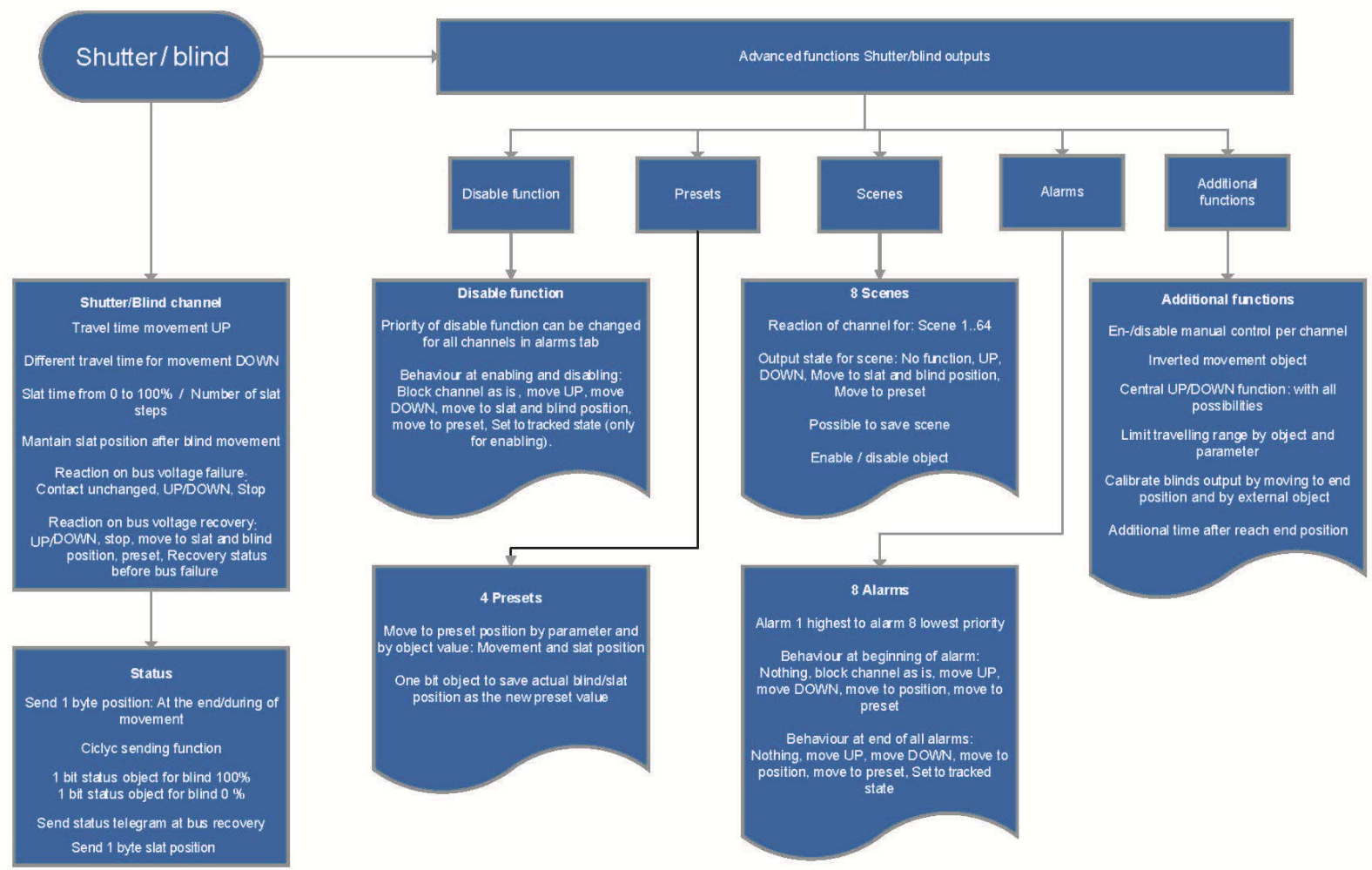
ANNEXES 2 FLOWCHARTS –

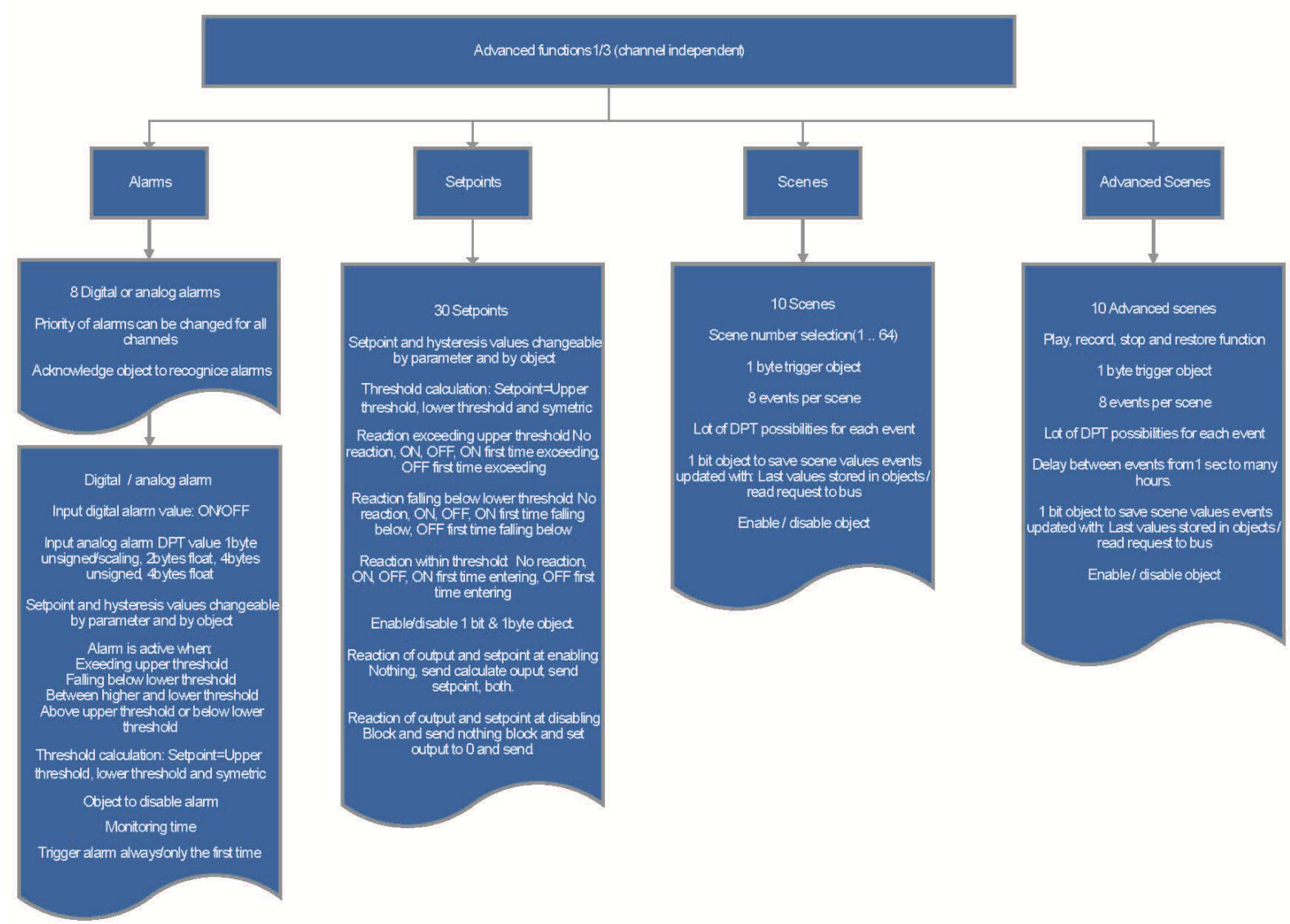

GENERAL SETTINGS
ALL INPUTS
ALL OUTPUTS
ADV. FUNCTIONS
Binary input
Movement detector
Temperature sensor
Binary output
Shutter/Blind
Analog & digital alarms
Logic functions
Scene controller
Advanced scene controller
Timers
Setpoints
Overwrite en-user parameters
Behaviour at bus recovery
Monitoring/ doubling inputs
2,7K Ohm end-line resistor
Open circuit alarm (N.C.)
Short circuit alarm (N.O.)
Both (open/short)
Acknowledge function
Arm/disarm function
Switching value
Dimming
Shutter
KNX Scene
Multiple operations
Flashing
Sequence
Counter
Lighting channel
HVAC Channel
Sensor physical input
Sensor object input
Weighted output value
Bus failure
Central ON/OFF
Counters
Scenes
Timers
Alarms
Disable function
Manual control
Bus failure
Scenes
Presets
Alarms
Disable function
Manual control

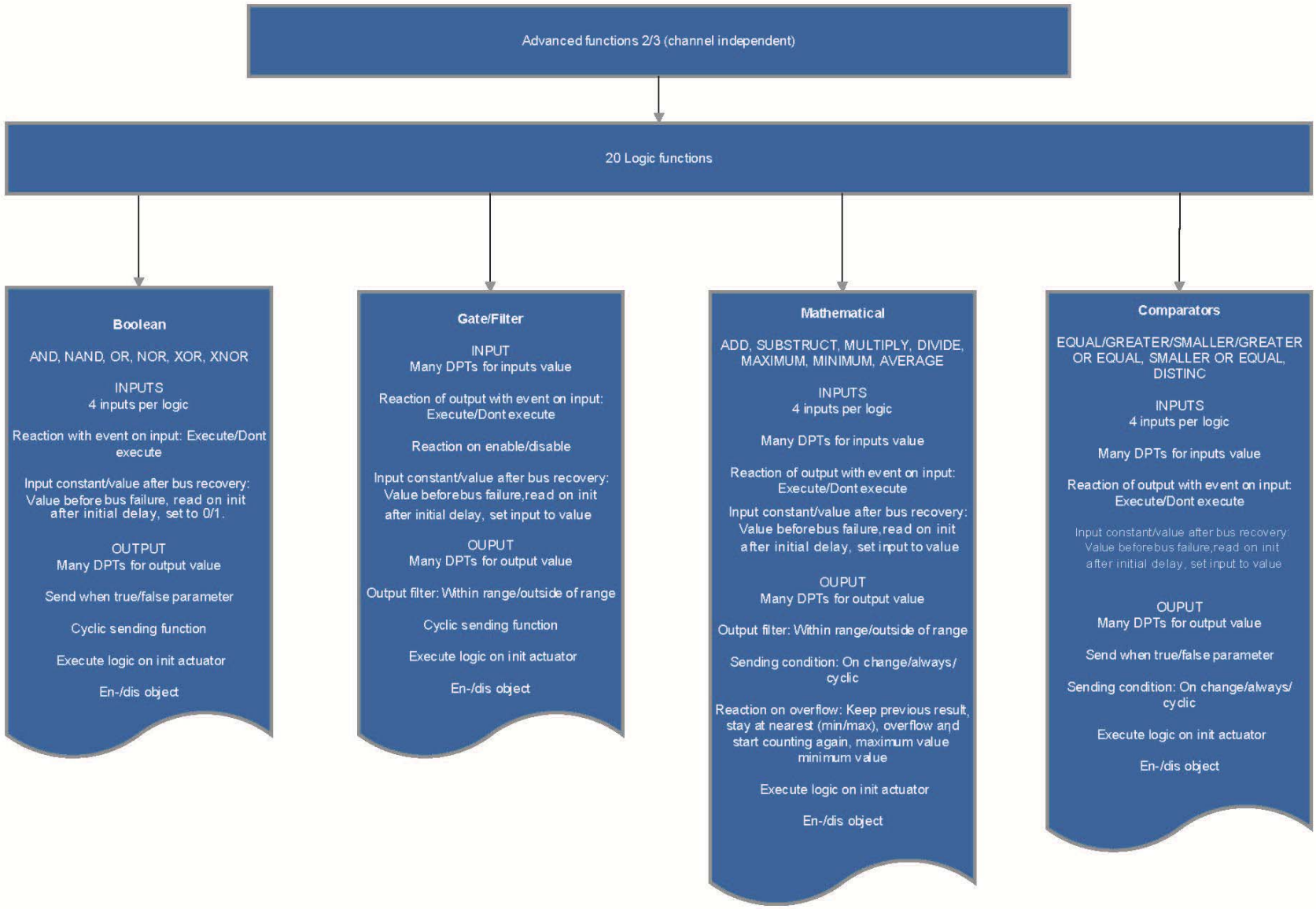

Binary input
Switching value
Dimming
Shutter
KNX Scene
Multiple operations
Flashing
Sequence
Counter
Switching value
Event for Short / Long operation & opening / closing contact: ON/OFF/Toggle/Value
Datapoint types:
1bit, 1byte unsigned, 1byte scaling (%), 2bytes float, 4bytes unsigned, 4 bytes float
Delays
Start stop cyclic sending (Closing /Opening / Both)
Monitor input with 2,7kohm end-line resistor
Short circuit, Open circuit and Both
KNX Scenes
Scene 1..64
Short: Play (0..63), Long: Save (128.. 191)
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input
5 Multiple operations
Action on defined pulse (1..10)
Short/Long operation
Datapoint types:
1bit, 1byte unsigned, 1byte scaling (%), 2bytes float
Condition for sending value:
Only evaluates last executed pulse operation
OR
Evaluates immediately when operations = configured pulse
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input
Counter
Same as Upwards/Backwards counter in flowchart of binary output
+
Count on (Rising edge, Falling edge, Rising and falling edge)
Additional Upwards counter
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input
Dimming
On / Brighter, Off / Darker
Toggle Brighter/Darker
Dimming direction after switching ON
Cyclic sending
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input
Shutter
Event for Short /Long operation:
Stop/Step Up, Stop/Step Down,
Toggle Stop/Step
Slat time for stop telegram when releasing
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input
Flashing
Flashing on:
Close = flash, open = nothing
Close = nothing, open = flash
Close = flash, open = stop
Close = Stop, open = flash
Both = start flashing
Time ON, Time OFF
Number of repetitions
Stop flashing at disabling/enabling input
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input
Sequence
DPT: 1bit, 1byte unsigned, 1byte scaling (%), 2bytes float
Single sequence
Incremental Loop (1>2>3>4>Off>1>...)
Incremental (OFF>1>2>3>4>stay in 4)
Decremental Loop (4>3>2>1>OFF>4>...)
Decremental (4>3>2>1>stay in OFF)
Toggle pause (1>2>3>4>Off>1>...), (4>3>2>1>OFF>4>...)
Toggle (Off>1>2>3>4>3>2>1>Off>1>...)
Multiple sequence
Inc. ON Loop (1>1+2>1+2+3>1+2+3+4>All Off>1>...)
Inc. ON (OFF>1>1+2>1+2+3>1+2+3+4>stay in 4)
Dec. OFF Loop (4+3+2+1>3+2+1>2+1>1>OFF>4+3+2+1...)
Dec. OFF (4+3+2+1>3+2+1>2+1>1>stay in OFF)
Toggle pause (1>1+2>1+2+3>1+2+3+4>Off>1>...), (4+3+2+1>OFF>4>...)
Toggle (Off>1>2>3>4>3>2>1>Off>1>...)
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input

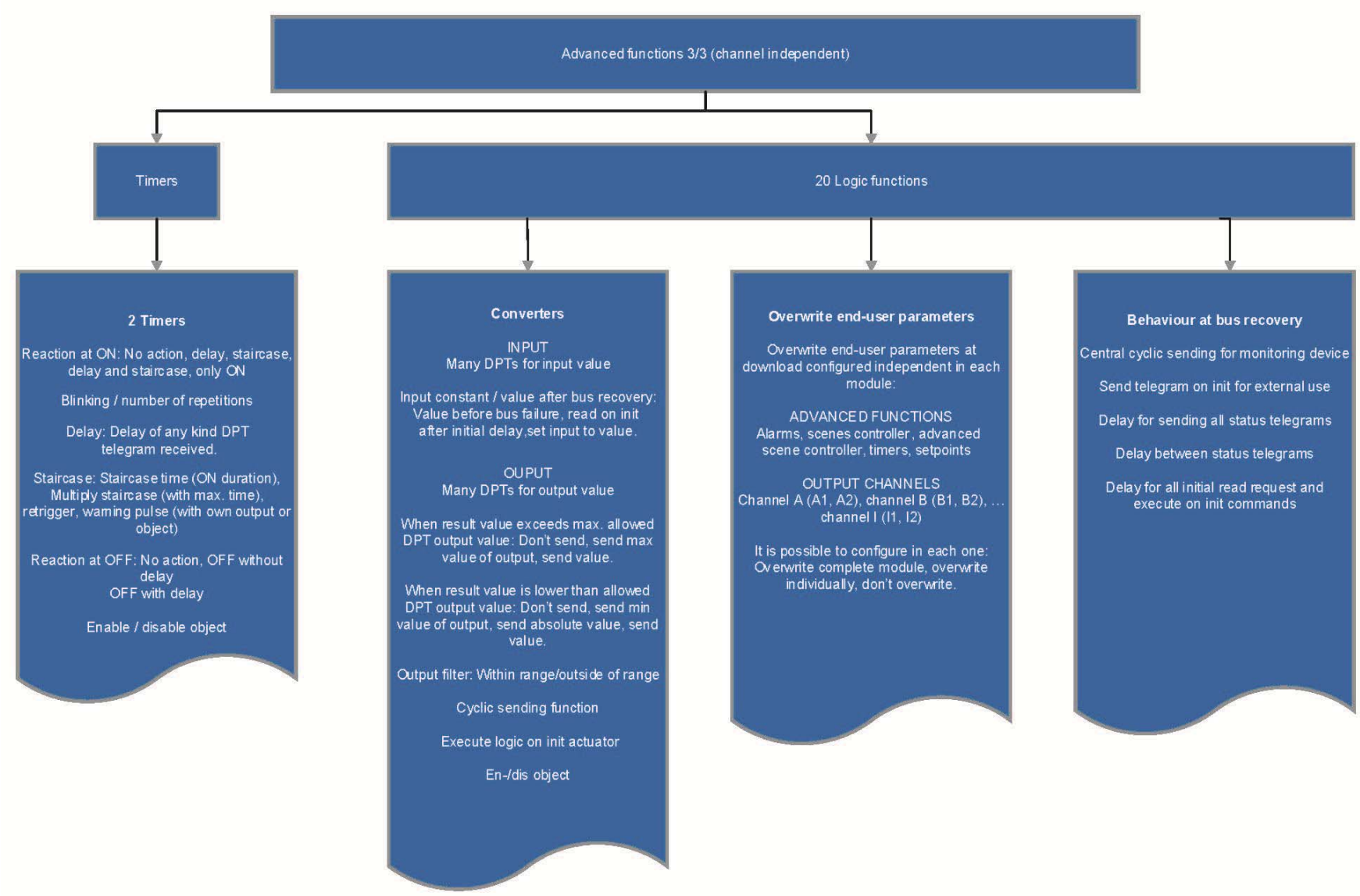

Movement detector
Detection time period can be set per parameter or in external detector
Blocking time after end of detection
Lighting channel
HVAC channel
Lighting channel
Events (any value & also cyclical) for beginning & end of detection
Datapoint types: 1bit, 1byte unsigned, 1byte scaling (%), 2bytes float 4 bytes unsigned, 4 bytes float
Brightness depended switching with external object
2 Enable / disable functions permitting continuous ON & continuous OFF lighting
HVAC channel
Initial waiting time for HVAC activation
Events (any value & also cyclical) for beginning & end of detection
Datapoint types: 1bit, 1byte unsigned, 1byte scaling (%), 2bytes float 4 bytes unsigned, 4 bytes float
Enable / disable function
Temperature sensor
Temperature sensor
2 Sources configurable: connected sensor & external object
Weighted source % (first- second)
Datapoint types: 2 bytes & 4 bytes float
Sensor calibration
Sending condition
Always, never, on change, cyclic sending, send with changes higher than (°C)
Cyclic supervision of temperature input
If value is not received within supervision time
- Only use other sensor (without weight)
- Use last value
Send all status telegrams after bus recovery


Binary outputs
Advanced functions binary outputs
Disable function
Timers
Scenes
Alarms
Additional functions
Counters
Binary channel
Type of contact: NO - Normally Open NC - Normally Closed
Reaction on bus voltage failure & recovery: Contact unchanged, ON, OFF, Status before bus failure, Timer 1 reaction at ON or OFF
Status
Sending condition: Only on change, always, only on change – inverted, always – Inverted, Only reading
Ciclyc sending & Delay function
Send status telegram at bus recovery
Disable function
Priority of disable function can be changed for all channels in alarms tab
Behaviour at enabling and disabling: Leave channel as is, ON, OFF, timer 1 reaction at ON or OFF, Set to tracked state
2 Timers
Reaction at ON: No action, delay, staircase, delay and staircase, only ON
Blinking / number of repetitions
Staircase: Staircase time (ON duration), Multiply staircase (with max. time), retrigger, warning pulse (with own output or object)
Reaction at OFF: No action, OFF without delay
OFF with delay
Enable / disable object
Reaction when SWITCHING or SCENE objects receive a value while timer is active
8 Scenes
Reaction of channel for: Scene 1..64
Output state for scene: No function, ON, OFF
Timer 1 reaction at ON or OFF
Possible to save scene
Enable / disable object
8 Alarms
Alarm 1 highest to alarm 8 lowest priority
Behaviour at beginning of alarm: Nothing, block channel as is, ON, OFF, timer 1 reaction at ON or OFF
Behaviour at end of all alarms: Nothing, ON, OFF, timer 1 reaction at ON or OFF, Set to tracked state
Additional functions
En-/disable manual control per channel
Central ON/OFF function: with all possibilities
Additional object: Inverted, toggle only with 0, toggle only with 1, toggle with 0 and 1
Counters
2 simultaneous active counters per channel:
Run hour counter/Switching counter
2 possibilities for each counter: Upward or Backward
Additional functions:
Threshold value
Object for updating threshold value
Reaction on overflow
Conversion factor
Additional object to store last value of counter on reset


Shutter / blind
Advanced functions Shutter/blind outputs
Disable function
Presets
Scenes
Alarms
Additional functions
Shutter/Blind channel
Travel time movement UP
Different travel time for movement DOWN
Slat time from 0 to 100% / Number of slat steps
Mantain slat position after blind movement
Reaction on bus voltage failure: Contact unchanged, UP/DOWN, Stop
Reaction on bus voltage recovery: UP/DOWN, stop, move to slat and blind position, preset, Recovery status before bus failure
Status
Send 1 byte position: At the end/during of movement
Ciclyc sending function
1 bit status object for blind 100%
1 bit status object for blind 0 %
Send status telegram at bus recovery
Send 1 byte slat position
Disable function
Priority of disable function can be changed for all channels in alarms tab
Behaviour at enabling and disabling: Block channel as is , move UP, move DOWN, move to slat and blind position, move to preset, Set to tracked state (only for enabling).
4 Presets
Move to preset position by parameter and by object value: Movement and slat position
One bit object to save actual blind/slat position as the new preset value
8 Scenes
Reaction of channel for: Scene 1..64
Output state for scene: No function, UP, DOWN, Move to slat and blind position, Move to preset
Possible to save scene
Enable / disable object
8 Alarms
Alarm 1 highest to alarm 8 lowest priority
Behaviour at beginning of alarm: Nothing, block channel as is, move UP, move DOWN, move to position, move to preset
Behaviour at end of all alarms: Nothing, move UP, move DOWN, move to position, move to preset, Set to tracked state
Additional functions
En-/disable manual control per channel
Inverted movement object
Central UP/DOWN function: with all possibilities
Limit travelling range by object and parameter
Calibrate blinds output by moving to end position and by external object
Additional time after reach end position


Advanced functions 1/3 (channel independent)
Alarms
Setpoints
Scenes
Advanced Scenes
8 Digital or analog alarms
Priority of alarms can be changed for all channels
Acknowledge object to recognice alarms
Digital / analog alarm
Input digital alarm value: ON/OFF
Input analog alarm DPT value 1byte unsigned/scaling, 2bytes float, 4bytes unsigned, 4bytes float
Setpoint and hysteresis values changeable by parameter and by object
Alarm is active when:
Exeeding upper threshold
Falling below lower threshold
Between higher and lower threshold
Above upper threshold or below lower threshold
Threshold calculation: Setpoint=Upper threshold, lower threshold and symetric
Object to disable alarm
Monitoring time
Trigger alarm always/only the first time
30 Setpoints
Setpoint and hysteresis values changeable by parameter and by object
Threshold calculation: Setpoint=Upper threshold, lower threshold and symetric
Reaction exceeding upper threshold No reaction, ON, OFF, ON first time exceeding, OFF first time exceeding
Reaction falling below lower threshold No reaction, ON, OFF, ON first time falling below, OFF first time falling below
Reaction within threshold No reaction, ON, OFF, ON first time entering, OFF first time entering
Enable/disable 1 bit & 1byte object
Reaction of output and setpoint at enabling Nothing, send calculate ouput, send setpoint, both.
Reaction of output and setpoint at disabling Block and send nothing block and set output to 0 and send
10 Scenes
Scene number selection(1 .. 64)
1 byte trigger object
8 events per scene
Lot of DPT possibilities for each event
1 bit object to save scene values events updated with: Last values stored in objects / read request to bus
Enable / disable object
10 Advanced scenes
Play, record, stop and restore function
1 byte trigger object
8 events per scene
Lot of DPT possibilities for each event
Delay between events from 1 sec to many hours.
1 bit object to save scene values events updated with: Last values stored in objects / read request to bus
Enable / disable object


Advanced functions 2/3 (channel independent)
20 Logic functions
Boolean
AND, NAND, OR, NOR, XOR, XNOR
INPUTS
4 inputs per logic
Reaction with event on input: Execute/Dont execute
Input constant/value after bus recovery: Value before bus failure, read on init after initial delay, set to 0/1.
OUTPUT
Many DPTs for output value
Send when true/false parameter
Cyclic sending function
Execute logic on init actuator
En-/dis object
Gate/Filter
INPUT
Many DPTs for inputs value
Reaction of output with event on input: Execute/Dont execute
Reaction on enable/disable
Input constant/value after bus recovery: Value beforebus failure,read on init after initial delay, set input to value
OUPUT
Many DPTs for output value
Output filter: Within range/outside of range
Cyclic sending function
Execute logic on init actuator
En-/dis object
Mathematical
ADD, SUBSTRUCT, MULTIPLY, DIVIDE, MAXIMUM, MINIMUM, AVERAGE
INPUTS
4 inputs per logic
Many DPTs for inputs value
Reaction of output with event on input: Execute/Dont execute
Input constant/value after bus recovery: Value beforebus failure,read on init after initial delay, set input to value
OUPUT
Many DPTs for output value
Output filter: Within range/outside of range
Sending condition: On change/always/ cyclic
Reaction on overflow: Keep previous result, stay at nearest (min/max), overflow and start counting again, maximum value minimum value
Execute logic on init actuator
En-/dis object
Comparators
EQUAL/GREATER/SMALLER/GREATER OR EQUAL, SMALLER OR EQUAL, DISTINC
INPUTS
4 inputs per logic
Many DPTs for inputs value
Reaction of output with event on input: Execute/Dont execute
Input constant/value after bus recovery: Value beforebus failure,read on init after initial delay, set input to value
OUPUT
Many DPTs for output value
Send when true/false parameter
Sending condition: On change/always/ cyclic
Execute logic on init actuator
En-/dis object


Advanced functions 3/3 (channel independent)
Timers
20 Logic functions
2 Timers
Reaction at ON: No action, delay, staircase, delay and staircase, only ON
Blinking / number of repetitions
Delay: Delay of any kind DPT telegram received.
Staircase: Staircase time (ON duration), Multiply staircase (with max. time), retrigger, warning pulse (with own output or object)
Reaction at OFF: No action, OFF without delay
OFF with delay
Enable / disable object
Converters
INPUT
Many DPTs for input value
Input constant / value after bus recovery: Value before bus failure, read on init after initial delay,set input to value.
OUPUT
Many DPTs for output value
When result value exceeds max. allowed DPT output value: Don't send, send max value of output, send value.
When result value is lower than allowed DPT output value: Don't send, send min value of output, send absolute value, send value.
Output filter: Within range/outside of range
Cyclic sending function
Execute logic on init actuator
En-/dis object
Overwrite end-user parameters
Overwrite end-user parameters at download configured independent in each module:
ADVANCED FUNCTIONS
Alarms, scenes controller, advanced scene controller, timers, setpoints
OUTPUT CHANNELS
Channel A (A1, A2), channel B (B1, B2), … channel I (I1, I2)
It is possible to configure in each one: Overwrite complete module, overwrite individually, don't overwrite.
Behaviour at bus recovery
Central cyclic sending for monitoring device
Send telegram on init for external use
Delay for sending all status telegrams
Delay between status telegrams
Delay for all initial read request and execute on init commands