PowerBlock

o8/o16 Multi Series

Version 1.0.0

Application program description

Contents

1 INTRODUCTION 4

1.1 GENERAL PRODUCT INFORMATION 4

1.2 PRELIMINARY BASIC CONCEPTS 4

2 ETS COMMUNICATION OBJECTS OVERVIEW 6

3 PARAMETER PAGE: GENERAL SETTINGS 35

4 PARAMETER PAGE: OUTPUTS 37

4.1 CHANNEL A1…X1 (BINARY) 38

4.1.1 Status 39

4.1.2 Advanced Functions 40

4.2 CHANNEL X1 (SHUTTER / BLIND) 62

4.2.1 SLAT PARAMETERS 64

4.2.2 Bus failure 66

4.2.3 Advanced functions 68

4.2.4 Status shutter 81

5 PARAMETER PAGE: FAN COIL 83

5.1 FAN COIL SETTINGS 83

5.1.1 Manual device buttons 85

5.1.2 Behaviour at bus failure/recovery 85

5.1.3 Special operation mode 1(Tab fan speed) 87

5.1.4 Operation mode (Valve tab) 89

5.1.5 Scenes 1..4 90

5.1.6 Alarms fan (Fan tab) 92

5.2 FAN SPEED 93

5.3 FAN AUTO 96

5.4 FAN MANUAL 98

5.5 VALVE 101

5.5.1 Alarms valve (Valve tab) 104

5.6 STATUS 105

6 PARAMETER PAGE: ADVANCED FUNCTIONS 108

6.1 ALARMS 108

6.1.1 Alarm 1…8 110

6.1.2 Digital 111

6.1.3 Analog 111
6.2 LOGICS 113
6.2.1 Boolean 113
6.2.2 Gate / Filter 116
6.2.3 Mathematical 117
6.2.4 Comparators 119
6.2.5 Converters 121
6.3 SCENE CONTROLLER 124
6.3.1 First scene / Tenth scene 124
6.4 TIMERS 126
6.4.1 Timer 1 / Timer 10 126
6.5 SETPOINTS 131
6.5.1 Setpoints Tab 131
6.5.2 Setpoints 1 … 3 131
6.5.3 Setpoints 4 … 10 135
6.6 FACADE CONTROL 139
6.6.1 Facade 1..4 139
6.7 INTERNAL VARIABLES 142
6.7.1 Variables 1…10 142
6.8 OVERWRITE END-USER PARAMETER VALUES AT DOWNLOAD 150
6.9 ENDUSER PARAMETERS 150
6.9.1 ADVANCED FUNCTIONS 150
6.10 CENTRAL SENDING OBJECT FOR MONITORING DEVICE 153
6.11 BEHAVIOUR AT BUS RECOVERY 154
7 FIRMWARE VERSION AND UPDATE 155
8 RESET TO CONDITIONS AT DELIVERY 155
9 ANNEX 156
9.1 ANNEX 1: MANUAL CONTROL (PARAMETER MODE) 156
9.1.1 PARAMETER MODE 156
9.1.2 TEST MODE 157
9.2 ANNEX 2 FLOWCHART 158

1 Introduction

Using the application program

Product family:	Actuators
Product type:	Actuators
Manufacturer:	IPAS GmbH
Name:	Power Block o8 / o16 Multi actuator range
Order Nr.:	See following table

Product name	Order number
Power Block o8 Multi Power Block o16 Multi	77024-180-03 72130-180-06

The following describes the application based on the PowerBlock o8 Multi hardware. This application is identical to the device type PowerBlock o16 Multi hardware.

1.1 General product information

Installing the application program

The application for the Power Blocks o8/o16 Multi 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. It can be found under product family “Outputs“ and product type “Actuators“.

1.2 Preliminary basic concepts

Output: channel type selection
The outputs of the PowerBlock o8/o16 Multi are divided into 2 or 4 channels depending on the version. Each channel consists of 2 or 4 mechanical outputs (relays) with a total of 4 or 8 relays per group.
If the channel type is selected to be a“Binary/Shutter, then each of the 2 channels will be ready to be configured as 2 capacitive relay 140uF” outputs or as one “Shutter/Blind channel.

In the case of a “Capacitive relay 140uF” selection, it will have two fully independent outputs in the Application program per channel.

In the case of a “Binary/Shutter” selection, the first relay will be for movement UP and second one for movement DOWN.

If the channel type is selected to be a "Fan Coil", then these four outputs (2 channels) will work as one Fan Coil controller. I.e. for channel A & B Fan Coil selection, the relays will be assigned in the next order:

- Output A1: Fan Speed 1
- Output A2: Fan Speed 2
- Output B1: Fan Speed 3
- Output B2: PWM Valve

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”.

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.

2 ETS communication objects overview

The Power Block actuators communicate via the KNX bus based on powerful communication stacks. A total of 998 communication objects for the Power Block o8/o16 Multi are available for communication.

	Text	Function text	Object Size	Flags	Datapoint type
1	Central switching	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
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---	[1.001] DPT_Switch
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---	[1.001] DPT_Switch
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--	[[1.001] DPT_Switch
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 therefor the "Line status light" will switch OFF.					
4	Telegram at bus recovery	> Sends parameterized value	1 Bit	--CT--	[1.001] DPT_Switch
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---	[1.003] DPT_Enable
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---	[1.003] DPT_Enable
The manual buttons on the device can be deactivated by this object like this: Disable = 0 / Enable = 1					
7	Alarm 1	< On / Off	1 Bit	RWC--I	[1.001] DPT_Switch
This object is the alarm 1 trigger object. In the parameters one can define with which value it should be in the alarm state.					
7	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.					

7	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.					
7	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.					
7	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.					
7	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.					
7	Alarm ACK	< Ack. with 0	1 Bit	-WC---	[1.016] DPT_Acknowledge
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					
15	Alarm ACK	< Ack. with 1	1 Bit	-WC---	[1.016] DPT_Acknowledge
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					
16	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					
16	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					
16	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					
16	Alarm 1 setpoint	< 4 bytes unsigned	4 Bytes	RWC---	[12.1] DPT_Value_4_Ucount
If the alarm is configured to be an analog alarm then the threshold of this alarm can be set by this object					
16	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					
24	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					
24	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					
24	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					
24	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					
24	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					
32	Alarm 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1.003] DPT_Enable
The alarm can be disabled by sending a 1 to this object.					
40	Alarm 1 status	> ON = Alarm, OFF = No alarm	1 Bit	R-CT--	[1] 1.005 DPT_Alarm
This object will send the actual alarm status value					
48	Logic 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1.003] DPT_Enable
The logic function can be disabled by sending a 0					
48	Logic 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1.003] DPT_Enable
The logic function can be disabled by sending a 1					
49	Logic 1 input 1	< On / Off	1 Bit	RWCTU-	[1.001] DPT_Switch
This is the first of 4 logic inputs of this logic block					
49	Logic 1 input 1	< 0..100%	1 Byte	RWCTU-	[5.1] DPT_Scaling
This is the first of 4 logic inputs of this logic block					
49	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					
49	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					
49	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					
49	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					
49	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					
49	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					
49	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					

49	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					
48	Logic 1 input 2	< On / Off	1 Bit	RWCTU-	[1.001] DPT_Switch
This is the second of 4 logic inputs of this logic block					
50	Logic 1 Enable / Disable Gate	< Disable = 1 / Enable = 0	1 Bit	RWCT--	[1.003] DPT_Enable
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.					
50	Logic 1 Enable / Disable Gate	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1.003] DPT_Enable
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.					
50	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					
50	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					
50	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					
50	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					
50	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					
50	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					
50	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					
50	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					
50	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					
51	Logic 1 input 3	< On / Off	1 Bit	RWCTU-	[1.001] DPT_Switch
This is the third of 4 logic inputs of this logic block					
51	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					
51	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					
51	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					
51	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					
51	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					
51	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					
51	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					
51	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					
51	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					
52	Logic 1 input 4	< On / Off	1 Bit	RWCTU-	[1.001] DPT_Switch
This is the fourth of 4 logic inputs of this logic block					
52	Logic 1 input 4	< 0..100%	1 Byte	RWCTU-	[5.1] DPT_Scaling
This is the fourth of 4 logic inputs of this logic block					
52	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					
52	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					
52	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					
52	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					

52	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					
52	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					
52	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					
52	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					
53	Logic 1 output	> On / Off	1 Bit	R-CT--	[1.001] DPT_Switch
This is the output of this logic block and the DPT can differ the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
53	Logic 1 output	> 1 byte unsigned	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount
This is the output of this logic block and the DPT can differ the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
53	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 the input. The value when true or false or the result of the logic block will be sent with this object.					
358	Advanced Scene 1 input	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch

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.					
358	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.					
358	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.					
358	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.					
358	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.					
358	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.					
358	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.					
358	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.					
358	Advanced Scene 1 input	< 4 bytes signed	4 Bytes	-WC---	[13.1] DPT_Value_4_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.					
358	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.					
359	Advanced Scene 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1.003] DPT_Enable
The scene can be disable with a 1					
359	Advanced Scene 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1.003] DPT_Enable
The scene can be disable with a 0					
360	Advanced Scene 1 event 1	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the first event for the first advanced scene.					
360	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.					
360	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.					

360	Advanced Scene 1 event 1	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the first event for the first advanced scene.					
360	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.					
360	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.					
360	Advanced Scene 1 event 1	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the first event for the first advanced scene.					
360	Advanced Scene 1 event 1	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the first event for the first advanced scene.					
360	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.					
360	Advanced Scene 1 event 1	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the first event for the first advanced scene.					
361	Advanced Scene 1 event 2	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the second event for the first advanced scene.					
361	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.					
361	Advanced Scene 1 event 2	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the second event for the first advanced scene.					
361	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.					
361	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.					
361	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.					
361	Advanced Scene 1 event 2	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the second event for the first advanced scene.					
361	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.					
361	Advanced Scene 1 event 2	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the second event for the first advanced scene.					
361	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.					
361	Advanced Scene 1 event 3	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the third event for the first advanced scene.					
362	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.					
362	Advanced Scene 1 event 3	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the third event for the first advanced scene.					
362	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.					
362	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.					
362	Advanced Scene 1 event 3	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the third event for the first advanced scene.					
362	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.					
362	Advanced Scene 1 event 3	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the third event for the first advanced scene.					
362	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.					
362	Advanced Scene 1 event 3	<> 4 bytes unsigned	4 Bytes	-WCTU-	[12.1] DPT_Value_4_Ucount
This is the third event for the first advanced scene.					
363	Advanced Scene 1 event 4	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the fourth event for the first advanced scene.					
363	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.					

363	Advanced Scene 1 event 4	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fourth event for the first advanced scene.					
363	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.					
363	Advanced Scene 1 event 4	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the fourth event for the first advanced scene.					
363	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.					
363	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.					
363	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.					
363	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.					
363	Advanced Scene 1 event 4	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fourth event for the first advanced scene.					
364	Advanced Scene 1 event 5	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the fifth event for the first advanced scene.					
364	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.					
364	Advanced Scene 1 event 5	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fifth event for the first advanced scene.					
364	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.					
364	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.					
364	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.					
364	Advanced Scene 1 event 5	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx

This is the fifth event for the first advanced scene.					
364	Advanced Scene 1 event 5	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fifth event for the first advanced scene.					
364	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.					
364	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.					
365	Advanced Scene 1 event 6	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the sixth event for the first advanced scene.					
365	Advanced Scene 1 event 6	<> 1 byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the sixth event for the first advanced scene.					
365	Advanced Scene 1 event 6	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the sixth event for the first advanced scene.					
365	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.					
365	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.					
365	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.					
365	Advanced Scene 1 event 6	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the sixth event for the first advanced scene.					
365	Advanced Scene 1 event 6	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the sixth event for the first advanced scene.					
365	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.					
365	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.					
366	Advanced Scene 1 event 7	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the seventh event for the first advanced scene.					

366	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.					
366	Advanced Scene 1 event 7	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the seventh event for the first advanced scene.					
366	Advanced Scene 1 event 7	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the seventh event for the first advanced scene.					
366	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.					
366	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.					
366	Advanced Scene 1 event 7	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the seventh event for the first advanced scene.					
366	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.					
366	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.					
366	Advanced Scene 1 event 7	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the seventh event for the first advanced scene.					
367	Advanced Scene 1 event 8	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the eighth event for the first advanced scene.					
367	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.					
367	Advanced Scene 1 event 8	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the eighth event for the first advanced scene.					
367	Advanced Scene 1 event 8	<> 1byte unsigned	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount
This is the eighth event for the first advanced scene.					
367	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.					
367	Advanced Scene 1 event 8	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx

This is the eighth event for the first advanced scene.					
367	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.					
367	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.					
367	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.					
367	Advanced Scene 1 event 8	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the eighth event for the first advanced scene.					
458	Timer 1 trigger	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This is to trigger the first timer					
458	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)					
458	Timer 1 trigger	< 1 byte scaling	1 Byte	-WC---	[5.1] DPT_Scaling
This is to trigger the first timer (only for delay)					
458	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)					
458	Timer 1 trigger	< 2 bytes unsigned	2 Bytes	-WC---	[7.1] DPT_Value_2_Ucount
This is to trigger the first timer (only for delay)					
458	Timer 1 trigger	< 2 bytes float	2 Bytes	-WC---	[9] 9.xxx
This is to trigger the first timer (only for delay)					
458	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)					
458	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)					
458	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)					
458	Timer 1 trigger	< 4 bytes float	4 Bytes	-WC---	[14] 14.xxx
This is to trigger the first timer (only for delay)					

459	Timer 1 change factor/Remaining time	< 1 byte unsigned	1 Byte	RWCT--	[5.10] DPT_Value_1_Ucount
Change factor: 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.**Remaining time:** Additionally to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value. In order to disable this function, the "T" flag must be deactivated.					
460	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.					
461	Timer 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1.003] DPT_Enable
The timer can be disabled by this object by sending a 0					
462	Timer 1 output	> On / Off	1 Bit	--CT--	[1.1] DPT_Switch
This is the output object of the timer.					
462	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)					
462	Timer 1 output	> 1 byte unsigned	1 Byte	--CT--	[5.10] DPT_Value_1_Ucount
This is the output object of the timer. (only for the delay function)					
462	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)					
462	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)					
462	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)					
462	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)					
462	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)					
462	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) 462					
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)					
508	Setpoint 1 output value 1	> On / Off	1 Bit	R-CT--	[1.001] DPT_Switch
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					

509	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 an unblocking the setpoint					
509	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 an unblocking the setpoint					
509	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 an unblocking the setpoint					
509	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 an unblocking the setpoint					
509	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 an unblocking the setpoint					
509	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 an unblocking the setpoint					
510	Setpoint 1 Heat / Cool	< Heat = 1 / Cool = 0	1 Bit	RWC---	[1] 1.100
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)					
511	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					
511	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					
511	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					
511	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					
511	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					
511	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					
512	Setpoint 1 disable	< On / Off	1 Bit	RWC---	[1.003] DPT_Enable

The setpoint can be disabled with this object					
512	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.					
558	Facade 1 Blind position	< 1 byte scaling	1 Byte	-WC---	[5.001] DPT_Scaling
All the shutter/blind channels assigned to the Facade control group, can be positioned with this object. When Facade control is active, channel slats and blind position objects will be inactive.					
559	Facade 1 Slat position	< 1 byte scaling	1 Byte	-WC---	[5.001] DPT_Scaling
All the slat blind channels assigned to the Facade control group, can be positioned with this object. When Facade control is active, channel slats and blind position objects will be inactive.					
560	Facade 1 Auto / Manual_Temporized	< 1=Facade / 0=Manual Temp.	1 Bit	-WC---	[1.1] DPT_Switch
The Facade control mode can be deactivated temporally when this communication object receives the value 0. At the end of the temporization, the slat/blind channel objects will be inactive again. For cancelling the temporization, the communication object must receive the value 1					
560	Facade 1 Auto / Manual	< 1=Facade / 0=Manual	1 Bit	-WC---	[1.1] DPT_Switch
The Facade control mode can be deactivated when this communication object receives the value 0. For cancelling the Manual control, the communication object must receive the value 1, so the slat/blind channel objects will be inactive again					
561	Facade 1 Auto / Manual_Temp. status	> 1=Facade / 0=Manual Temp.	1 Bit	R-CT--	[1.1] DPT_Switch
This status object indicates if the Facade control or Manual temporization is active					
561	Facade 1 Auto / Manual status	> 1=Facade / 0=Manual	1 Bit	R-CT--	[1.1] DPT_Switch
This status object indicates if the Facade control or Manual mode is active					
574	Facade monitoring alarm	> ON = Alarm, OFF = No alarm	1 Bit	R-CT--	[1.005] DPT_Alarm
It is possible to supervise the received slat/blind position values in Facade control comm. objects from i.e a weather station. In case to don't receive any value during the parametrised time, this object alarm will be active.					
575	Facade Exclude Ch. A	< 0=No / 1= Exclude	1 Bit	-WC---	[1.1] DPT_Switch
It is possible to exclude only a unique channel from the Facade control group using this communication object.					
575	Facade Exclude Ch. A temporized	< 0=No / 1= Exclude Temp.	1 Bit	-WC---	[1.1] DPT_Switch
It is possible to exclude only a unique channel from the Facade control group temporary using this communication object, during the time established in the parameters.					
577	[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.					
577	[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					
578	[A1] Switching toggle/inverted	< Inverted	1 Bit	-WC---	[1.1] DPT_Switch

<table>
<tr><td colspan="6">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, when 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</td></tr>
<tr><td>578</td><td>[A] Stop (Blind=Stop/step)</td><td>< 0=stop/step, 1=stop/step</td><td>1 Bit</td><td>-WC---</td><td>[1.007] DPT_Step</td></tr>
<tr><td colspan="6">This is to stop/step the blind 0=stop/step up, 1=stop/step down</td></tr>
<tr><td>578</td><td>[A1] Switching toggle/inverted</td><td>< Toggle only with 0</td><td>1 Bit</td><td>-WC---</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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, when 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</td></tr>
<tr><td>578</td><td>[A1] Switching toggle/inverted</td><td>< Toggle with 0 and 1</td><td>1 Bit</td><td>-WC---</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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, when 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</td></tr>
<tr><td>578</td><td>[A1] Switching toggle/inverted</td><td>< Toggle only with 1</td><td>1 Bit</td><td>-WC---</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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, when 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</td></tr>
<tr><td>579</td><td>[A1] Switching status</td><td>> On / Off</td><td>1 Bit</td><td>R-CT--</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">This is the current status of the channel. The sending behaviour can be changed by the parameters</td></tr>
<tr><td>579</td><td>[A] Move to position</td><td>< 0..100%</td><td>1 Byte</td><td>-WC---</td><td>[5.1] DPT_Scaling</td></tr>
<tr><td colspan="6">The blind can be moved to a specific absolute position with this object.</td></tr>
<tr><td>580</td><td>[A] Move slat</td><td>< 0..100%</td><td>1 Byte</td><td>-WC---</td><td>[5.1] DPT_Scaling</td></tr>
<tr><td colspan="6">This object is to move the slats to an absolute position.</td></tr>
<tr><td>580</td><td>[A] Move slit</td><td>< 0..100%</td><td>1 Byte</td><td>-WC---</td><td>[5.1] DPT_Scaling</td></tr>
<tr><td colspan="6">This object is to move the slits to an absolute position.
100% value will close completely the shutter / 0% value will move the shutter to the bottom position but with all the slits in open position.</td></tr>
<tr><td colspan="6">The accumulated ON time of the channel is called the runhours and it is send 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.</td></tr>
<tr><td>580</td><td>[A1] RunHour counter value</td><td>> 4 bytes signed</td><td>4 Bytes</td><td>R-CT--</td><td>[13.100] DPT_time_lag_(s)</td></tr>
<tr><td colspan="6">The accumulated ON time of the channel is called the runhours and it is send 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.</td></tr>
<tr><td>581</td><td>[A] Change upper limit</td><td><> 0..100%</td><td>1 Byte</td><td>RWCT--</td><td>[5.1] DPT_Scaling</td></tr>
<tr><td colspan="6">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 send to the bus.</td></tr>
<tr><td>581</td><td>[A1] RunHour counter threshold</td><td>< Reading/writing threshold</td><td>4 Bytes signed</td><td>RWCT--</td><td>[13.100] DPT_time_lag_(s)</td></tr>
<tr><td colspan="6">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.</td></tr>
</table>

581	[A1] RunHour counter threshold	< Reading threshold	4 Bytes signed	R-CT--	[13.100] DPT_time_lag_(s)
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.					
582	[A1] RunHour counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	[1.005] DPT_Alarm
When crossing the threshold value the threshold alarm object will send an alarm message.					
582	[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 send to the bus.					
583	[A1] RunHour counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1.015] DPT_Reset
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 if the counter object should maintain and send the last value at reset					
583	[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.					
584	[A] Status blind lower end position	> 1 = Totally down / 0 = not	1 Bit	R-CT--	[1.001] DPT_Switch
When reaching the lower end position this object will send a 1, for any other position this object will be 0.					
584	[A1] RunHour counter value at reset	> 4 bytes signed	4 Bytes	R-CT--	[13.100] DPT_time_lag_(s)
In the parameters one can decide to activate this object should store and send the last value of the runhour counter at reset.					
585	[A] Status blind upper end position	> 1 = Totally up / 0 = not	1 Bit	R-CT--	[1.001] DPT_Switch
When reaching the upper end position this object will send a 1, for any other position this object will be 0.					
585	[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 in switches ON, OFF or both can be configured in the parameters					
585	[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 in switches ON, OFF or both can be configured in the parameters					
585	[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 in switches ON, OFF or both can be configured in the parameters					
586	[A] Status slit position	> 0..100%	1 Byte	R-CT--	[5.1] DPT_Scaling
This sends the status of the slit position after each movement.					
586	[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.					
586	[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.					

586	[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.					
586	[A1] Switching counter threshold	< Reading threshold	2 Bytes	R-CT--	[7.1] DPT_Value_2_Ucount
This object is to only read the threshold value.					
586	[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.					
586	[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.					
586	[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.					
587	[A] Preset 1 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
With a 1 this preset will be executed. 0 = No reaction					
587	[A1] Switching counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	[1.005] DPT_Alarm
When crossing the threshold value the threshold alarm object will send an alarm message.					
588	[A] Preset 2 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
With a 1 this preset will be executed. 0 = No reaction					
588	[A1] Switching counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1.015] DPT_Reset
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 if the counter object should maintain and send the last value at reset					
589	[A] Preset 3 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
With a 1 this preset will be executed. 0 = No reaction					
589	[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.					
589	[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.					
589	[A1] 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.					
590	[A] Preset 4 execute	< 1 = Execute, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
With a 1 this preset will be executed. 0 = No reaction					
590	[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.					
591	[A1] Scene disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1.003] DPT_Enable
The scene function for this channel can be disabled by sending a 1 to this object					
591	[A1] Scene disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1.003] DPT_Enable
The scene function for this channel can be disabled by sending a 0 to this object					
591	[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					
592	[A1] Timer 1 trigger	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This is to trigger the first timer associated to the channel					
592	[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					
593	[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					
593	[A1] Timer 1 change factor/Remaining time	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
Change factor: 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. **Remaining time:** Additionally to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value. In order to disable this function, the "T" flag must be deactivated.					
594	[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.					
594	[A] Preset 4 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 4					
595	[A1] Timer 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1.003] DPT_Enable
With this object the timer will be disabled by receiving a 0					
595	[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					
596	[A1] Timer 2 trigger	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This is to trigger the second timer associated to the channel					
596	[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					
597	[A] Preset 3 change slat position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling

<table>
<tr><td colspan="6">This is to change the blind absolute slat position which will be set when calling preset 3</td></tr>
<tr><td>597</td><td>[A1] Timer 2 change factor/Remaining time</td><td>< 1 byte unsigned</td><td>1 Byte</td><td>RWC---</td><td>[5.10] DPT_Value_1_Ucount</td></tr>
<tr><td colspan="6"><b>Change factor:</b> 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. <b>Remaining time:</b> Additionally to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value. In order to disable this function, the “T” flag must be deactivated.</td></tr>
<tr><td>598</td><td>[A1] Timer 2 warning pulse</td><td>> On / Off</td><td>1 Bit</td><td>R-CT--</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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.</td></tr>
<tr><td>598</td><td>[A] Preset 4 change slat position</td><td>< 0..100%</td><td>1 Byte</td><td>RWC---</td><td>[5.1] DPT_Scaling</td></tr>
<tr><td colspan="6">This is to change the blind absolute slat position which will be set when calling preset 4</td></tr>
<tr><td>599</td><td>[A] Preset 1 save</td><td>< 1 = Save, 0 = Nothing</td><td>1 Bit</td><td>-WC---</td><td>[1.001] DPT_Switch</td></tr>
<tr><td colspan="6">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</td></tr>
<tr><td>599</td><td>[A1] Timer 2 disable</td><td>< Disable = 0 / Enable = 1</td><td>1 Bit</td><td>RWCT--</td><td>[1.003] DPT_Enable</td></tr>
<tr><td colspan="6">The timer can be disabled by this object by sending a 0</td></tr>
<tr><td>600</td><td>[A] Preset 2 save</td><td>< 1 = Save, 0 = Nothing</td><td>1 Bit</td><td>-WC---</td><td>[1.001] DPT_Switch</td></tr>
<tr><td colspan="6">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</td></tr>
<tr><td>600</td><td>[A1] Disable channel</td><td>< On / Off</td><td>1 Bit</td><td>RWCT--</td><td>[1.003] DPT_Enable</td></tr>
<tr><td colspan="6">The channel can be disabled by this object. In the parameters one can decide to disable with a 1 or a 0.</td></tr>
<tr><td>601</td><td>[A] Preset 3 save</td><td>< 1 = Save, 0 = Nothing</td><td>1 Bit</td><td>-WC---</td><td>[1.001] DPT_Switch</td></tr>
<tr><td colspan="6">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</td></tr>
<tr><td>601</td><td>[A2] Switching On / Off</td><td>< On / Off</td><td>1 Bit</td><td>-WC---</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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.</td></tr>
<tr><td>602</td><td>[A2] Switching toggle/inverted</td><td>< Toggle only with 1</td><td>1 Bit</td><td>-WC---</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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, when configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output.</td></tr>
<tr><td>602</td><td>[A] Preset 4 save</td><td>< 1 = Save, 0 = Nothing</td><td>1 Bit</td><td>-WC---</td><td>[1.001] DPT_Switch</td></tr>
<tr><td colspan="6">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</td></tr>
<tr><td>602</td><td>[A2] Switching toggle/inverted</td><td>< Toggle with 0 and 1</td><td>1 Bit</td><td>-WC---</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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, when configured in the parameters to invert. But it can also be used to toggle the output regardless of the previous state of the output.</td></tr>
<tr><td>602</td><td>[A2] Switching toggle/inverted</td><td>< Toggle only with 0</td><td>1 Bit</td><td>-WC---</td><td>[1.1] DPT_Switch</td></tr>
<tr><td colspan="6">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, when 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</td></tr>
</table>

602	[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, when 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					
603	[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					
614	[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.					
615	[A] Scene disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1.003] DPT_Enable
The scene function for this channel can be disabled by sending a 1 to this object					
615	[A] Scene disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1.003] DPT_Enable
The scene function for this channel can be disabled by sending a 1 to this object					
604	[A2] RunHour counter value	> 4 bytes signed	4 Bytes	R-CT--	[13.100] DPT_time_lag_(s)
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.					
624	[A] Disable channel	< On / Off	1 Bit	RWCT--	[1.003] DPT_Enable
The channel can be disabled by this object. In the parameters one can decide to disable with a 1 or a 0.					
605	[A2] RunHour counter threshold	< Reading threshold	4 Bytes signed	R-CT--	[13.100] DPT_time_lag_(s)
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.					
605	[A2] RunHour counter threshold	< Reading/writing threshold	4 Bytes signed	RWCT--	[13.100] DPT_time_lag_(s)
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.					
606	[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					
606	[A2] RunHour counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	[1.005] DPT_Alarm
When crossing the threshold value the threshold alarm object will send an alarm message.					
607	[A] Disable limits / calibrate	< Disable =0 / En&calibrate =1	1 Bit	RWC---	[1.003] DPT_Enable
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.					
607	[A2] RunHour counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1.015] DPT_Reset
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 if the counter object should maintain and send the last value at reset					
608	[A2] RunHour counter value at reset	> 4 bytes signed	4 Bytes	R-CT--	[13.100] DPT_time_lag_(s)
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.					

609	[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 in switches ON, OFF or both can be configured in the parameters					
609	[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 in switches ON, OFF or both can be configured in the parameters					
609	[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 in switches ON, OFF or both can be configured in the parameters					
610	[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.					
610	[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.					
610	[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.					
610	[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.					
610	[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.					
610	[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.					
611	[A2] Switching counter alarm	> 1 = Alarm, 0 = No alarm	1 Bit	R-CT--	1.005] DPT_Alarm
When crossing the threshold value the threshold alarm object will send an alarm message.					
612	[A2] Switching counter reset	< 1 = Reset, 0 = Nothing	1 Bit	-WC---	[1.015] DPT_Reset
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 if the counter object should maintain and send the last value at reset					
613	[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.					
613	[A2] 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.					
613	[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.					
614	[A2] Scene number	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	-WC---	[18.001] DPT_Scene_control

With this object any of the configured scenes of this channel can be triggered and/or recorded.					
615	[A2] Scene disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1.003] DPT_Enable
The scene function for this channel can be disabled by sending a 1 to this object					
615	[A2] Scene disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1.003] DPT_Enable
The scene function for this channel can be disabled by sending a 0 to this object					
616	[A2] Timer 1 trigger	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This is to trigger the first timer					
617	Timer 1 change factor/Remaining time	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
Change factor: 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. **Remaining time:** Additionally to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value. In order to disable this function, the "T" flag must be deactivated..					
618	[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.					
616	[A2] Timer 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1.003] DPT_Enable
With this object the timer will be disabled by receiving a 0					
620	[A2] Timer 2 trigger	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This is to trigger the second timer					
621	[A2] Timer 1 change factor/Remaining time	< 1 byte unsigned	1 Byte	RWC---	[5.10] DPT_Value_1_Ucount
Change factor: 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. **Remaining time:** Additionally to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value. In order to disable this function, the "T" flag must be deactivated.					
622	[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.					
623	[A2] Timer 2 disable	< Disable = 0 / Enable = 1	1 Bit	RWCT--	[1.003] DPT_Enable
With this object the timer will be disabled by receiving a 0					
623	[A2] Disable channel	< On / Off	1 Bit	RWCT--	[1.003] DPT_Enable
The channel can be disabled by this object. In the parameters one can decide to disable with a 1 or a 0.					

FAN COIL MODULE

418	[FC1] On/Off	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
With this object the Fan Coil module will be switched ON/OFF					
419	[FC1] On/Off status	> On / Off	1 Bit	R-CT--	[1.001] DPT_Switch
The On/Off Fan Coil status telegram will be sent by this object					
420	[FC1] Heat / Cool mode	< 1=Heat/0 = Cool	1 Bit	-WC---	[1.100] DPT Cooling/heating
The Fan Coil heat/cool mode will be changed by this object					
421	[FC1] Heat / Cool status	> 1=Heat/0 = Cool	1 Bit	R-CT--	[1.100] DPT Cooling/heating
The heat/cool mode status telegram will be sent by this object					
422	[FC1] Heat / Cool PI control input	< 0..100%	1 byte	RWCT--	[5.001] Percentage (0..100%)
This object receives the PI Heat/Cool regulation value from the thermostat. It appears when parameter **“1 byte PI value (common Heat/Cool obj.)“** is selected in Valve -> Type of valve.					
423	[FC1] Heat PI control input	< 0..100%	1 byte	RWCT--	[5.001] Percentage (0..100%)
This object receives the PI Heat regulation value from the thermostat. It appears when parameter **“2 x 1 byte PI value (individual Heat/Cool obj.)”** is selected in Valve -> Type of valve.					
423	[FC1] Cool PI control input	< 0..100%	1 byte	RWCT--	[5.001] Percentage (0..100%)
This object receives the PI Cool regulation value from the thermostat. It appears when parameter **“2 x 1 byte PI value (individual Heat/Cool obj.)”** is selected in Valve -> Type of valve.					
424	[FC1] Heat / Cool mode control input	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This object receives the PWM Heat/Cool regulation value from the thermostat. It appears when parameter **“1 bit PWM value (common Heat/Cool obj.)”** is selected in Valve -> Type of valve.					
424	[FC1] Heat mode control input	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This object receives the PWM Heat regulation value from the thermostat. It appears when parameter “**2 x 1 bit PWM value (individual Heat/Cool obj.)”** is selected in Valve -> Type of valve					
424	[FC1] Cool mode control input	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This object receives the PWM Cool regulation value from the thermostat. It appears when parameter **“2 x 1 bit PWM value (individual Heat/Cool obj.)** is selected in Valve -> Type of valve					
425	[FC1] Heat / Cool Fan continuous control	< 0..100%	1 byte	-WC---	[5.001] Percentage (0..100%)
This object receives the PI Heat/Cool regulation value from the thermostat in order to control de Fan Speed. It appears when parameter **“1 bit PWM value (common Heat/Cool obj.)”** is selected in Valve -> Type of valve.					
425	[FC1] Cool Fan continuous control	< 0..100%	1 byte	-WC---	[5.001] Percentage (0..100%)
This object receives the PI Cool regulation value from the thermostat in order to control de Fan Speed. It appears when parameter **“2x 1 bit PWM value (individual Heat/Cool obj.)”** is selected in Valve -> Type of valve					
425	[FC1] Heat Fan continuous control	< 0..100%	1 byte	-WC---	[5.001] Percentage (0..100%)
This object receives the PI Heat regulation value from the thermostat in order to control de Fan Speed. It appears when parameter **“2x 1 bit PWM value (individual Heat/Cool obj.)”** is selected in Valve -> Type of valve					
426	[FC1] Cool control valve status (1 bit)	> On / Off	1 Bit	R-CT-	[1.001] DPT_Switch
The 1 bit output cooling valve status will be sent by this object					

427	[FC1] Heat control valve status (1 bit)	> On / Off	1 Bit	R-CT-	[1.001] DPT_Switch
The 1 bit output heating valve status will be sent by this object					
428	[FC1] Cool control valve status (1 byte)	> 0..100%	1 Byte	R-CT-	[5.001] Percentage (0..100%)
The 1 byte output cooling valve status will be sent by this object					
429	[FC1] Heat control valve status (1 byte)	> 0..100%	1 Byte	R-CT-	[5.001] Percentage (0..100%)
The 1 byte output heating valve status will be sent by this object					
430	[FC1] Scene disable	< Disable=0 / Enable = 1	1 Bit	-WC---	[1.003] DPT_Enable
With this object the scenes will be disabled when receiving a 0. When sending a 1 to this object the scenes will be enabled. The enable/disable values can be changed by parameter.					
431	[FC1] Scene 1	< Sc1 (0=Play 128=Rec)... Sc64	1 Byte	-WC---	[18.001] DPT_Scene_control
With this object any of the configured scenes of this FC1 can be triggered and/or recorded					
432	[FC1] Scene 1	< 1=Play Scene / 0=X	1 Bit	-WC---	[1.001] DPT_Switch
With this object any of the configured scenes of this FC1 can be triggered					
433	[FC1] Scene 1 Event 1 – On/Off	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
The On/Off value received in this object will be saved internally when the record function is activated. It appears when “Possible to save scene” parameter is selected.					
434	[FC1] Scene 1 Event 1 – Fan Speed	< 0=S0, 1=S1, 2=S2, 3=S3	1 Byte	-WC---	[5.010] DPT_Counter pulses (0..255)
The Fan Speed value received in this object will be saved internally when the record function is activated. It appears when “Possible to save scene” parameter is selected.					
435	[FC1] Scene 1 Event 2– On/Off	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
The On/Off value received in this object will be saved internally when the record function is activated. It appears when “Possible to save scene” parameter is selected.					
436	[FC1] Scene 1 Event 2– Fan Speed	< 0=S0, 1=S1, 2=S2, 3=S3	1 Byte	-WC---	[5.010] DPT_Counter pulses (0..255)
The Fan Speed value received in this object will be saved internally when the record function is activated. It appears when “Possible to save scene” parameter is selected.					
437	[FC1] Scene 1 Event 3– On/Off	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
The On/Off value received in this object will be saved internally when the record function is activated. It appears when “Possible to save scene” parameter is selected.					
438	[FC1] Scene 1 Event 3– Fan Speed	< 0=S0, 1=S1, 2=S2, 3=S3	1 Byte	-WC---	[5.010] DPT_Counter pulses (0..255)
The Fan Speed value received in this object will be saved internally when the record function is activated. It appears when “Possible to save scene” parameter is selected.					
463	[FC1] Day / Night	< 1=Day / Night=0	1 bit	-WC---	[1.022] DTP_Scene
With this object the Day scene can be activated when receiving a 1. When sending a 0 to this object the Night scene will be activated. The activation values and the assigned scenes can be changed by parameter.					
464	[FC1] Thermostat monitoring error	> 1=Error/0=Ok	1 bit	R-CT-	[1.005] DPT_Alarm
In case the thermostat stops sending the control values (PI or PWM) within the configured period time, this object will send an error with the value 1. When the thermostat starts to send the control values again, a value 0 will be sent.					

465	[FC1] Additional ventilation	< Disable=0/Enable=1	1 bit	RWC---	[1.003] DPT_Enable
With this object the Additional Ventilation function will be disabled when receiving a 0. When sending a 1 to this object the Additional Ventilation will be enabled.					
466	[FC1] Filter remaining time	< 4 bytes (Time(s))	4 bytes	R-CT-	[13.100] DPT_Time lag
This object sends periodically the remaining time for cleaning the Fan Coils filters.					
467	[FC1] Filter remaining time alarm	> 1=Alarm / 0=No alarm	1 bit	R-CT-	[1.005] DPT_Alarm
This object will send an alarm with value 1 when the “[FC1] Filter remaining time” object reaches 0 value. When the remaining time is restarted, a 0 value is sent resetting the previously alarm.					
468	[FC1] Filter remaining time reset	< 1=Reset / 0=Nothing	1 bit	-WC---	[1.015] DPT_Reset
With this object the filter remaining time will be restablished when receiving a value 1.					
469	[FC1] Operation mode 1	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
With this object, the operation mode 1 will be activated when receiving a 1. When sending a 0 to this object the operation mode 1 will be inactive. The opposite values are possible by changing it by parameters.					
470	[FC1] Operation mode 2	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
With this object, the operation mode 2 will be activated when receiving a 1. When sending a 0 to this object the operation mode 2 will be inactive. The opposite values are possible by changing it by parameters.					
471	[FC1] Operation mode 3	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
With this object, the operation mode 3 will be activated when receiving a 1. When sending a 0 to this object the operation mode 3 will be inactive. The opposite values are possible by changing it by parameters.					
472	[FC1] Operation mode 4	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
With this object, the operation mode 4 will be activated when receiving a 1. When sending a 0 to this object the operation mode 4 will be inactive. The opposite values are possible by changing it by parameters.					
473	[FC1] Operation mode	< 0=Exit, 1=M1, 2=M2, 3=M3, 4=M4	1 byte	-WC---	[5.010] DPT_Counter pulses (0..255)
With this object the different operation modes can be activated when receiving the corresponding value from 1 to 4. When the 0 value is received, the actual operation mode will be inactive.					
474	[FC1] Operation mode status (1 bit)	> On / Off	1 bit	R-CT-	[1.001] DPT_Switch
This object will send the status value 1 when an operation mode is active.					
475	[FC1] Operation mode status (1 byte)	< 0=Exit, 1=M1, 2=M2, 3=M3, 4=M4	1 byte	R-CT-	[5.010] DPT_Counter pulses (0..255)
This object will send the status value from 1 to 4 corresponding to the active operation mode or the value 0 when no operation modes are active.					
476	[FC1] Current temperature	2 byte floating point	2 byte	-WC---	[7.1] DPT_Value_2_Ucount
This object sends sends the current temperature.					
477	[FC1] Setpoint temperature	2 byte floating point	2 byte	-WC---	[7.1] DPT_Value_2_Ucount
This object sends sends the set temperature.					
478	[FC1] Auto / Manual	> 0 = Auto / 1 = Manual	1 bit	-WC---	[1.001] DPT_Switch
With this object the different operating mode Auto / Manual can be selected for the fan speed. The Automatic mode is active when a 0 is received, manual mode is active when a 1 value is received					

479	[FC1] Auto / Manual status	< 0 = Auto / 1 = Manual	1 bit	R-CT-	[1.001] DPT_Switch
This object will send the Auto/Manual status value					
480	[FC1] Disable timer to return to auto	> 1 = Stay in manual / 0 = Temporized	1 bit	-WC---	[1.001] DPT_Switch
With this object the manual fan control timer can be disable in order to avoid changing automatically to Auto mode after the parametrized time. Temporization will be active when receiving a 0 value and it will be disable when receiving a 1 value.					
481	[FC1] Fan speed 1	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object the Fan speed 1 will be active when 1 value is received. 0 value will do nothing. It appears when “**Yes, 3 x 1 bit**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
481	[FC1] Fan custom 1	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object 2 different parametrized fan behaviours can be active. When 1 value is received the associated behaviour to value 1 is active. When 0 value is received the associated behaviour to value 0 is active. It appears when “**Yes, custom**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
482	[FC1] Fan speed 2	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object the Fan speed 2 will be active when 1 value is received. 0 value will do nothing. It appears when “**Yes, 3 x 1 bit**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
482	[FC1] Fan custom 2	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object 2 different parametrized fan behaviours can be active. When 1 value is received the associated behaviour to value 1 is active. When 0 value is received the associated behaviour to value 0 is active. It appears when “**Yes, custom**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
483	[FC1] Fan speed 3	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object the Fan speed 13will be active when 1 value is received. 0 value will do nothing. It appears when “**Yes, 3 x 1 bit**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
483	[FC1] Fan custom 3	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object 2 different parametrized fan behaviours can be active. When 1 value is received the associated behaviour to value 1 is active. When 0 value is received the associated behaviour to value 0 is active. It appears when “**Yes, custom**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
484	[FC1] Fan custom 4	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object 2 different parametrized fan behaviours can be active. When 1 value is received the associated behaviour to value 1 is active. When 0 value is received the associated behaviour to value 0 is active. It appears when “**Yes, custom**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
485	[FC1] Fan custom 5	< 1 = On / 0 = Off	1 bit	-WC---	[1.001] DPT_Switch
With this object 2 different parametrized fan behaviours can be active. When 1 value is received the associated behaviour to value 1 is active. When 0 value is received the associated behaviour to value 0 is active. It appears when “**Yes, custom**” parameter is selected in “Fan manual” -> “Manual fan speed 1 bit objects”					
488	[FC1] Manual fan enumerated speed	< 0=S0; 1=S1; 2=S2; 3=S3	1 byte	-CWTU-	[5.010] DPT_Counter pulses (0..255)
With this object the different fan speeds can be changed when receiving the corresponding value from 0 to 3. 0 value will switch the fan OFF.					
492	[FC1] Fan speed enumerated status	< 0=S0; 1=S1; 2=S2; 3=S3	1 byte	CR-T--	[5.010] DPT_Counter pulses (0..255)
This object will send the status values from 0 to 3 corresponding to the active fan speed.					

493	[FC1] Fan speed scaling status	< 0%=S0;
		33%=S1;
		66%=S2;
		100%=S3
			1 byte	CR-T-	[5.001] Percentage (0..100%)

This object will send the status values from 0 to 3 corresponding to the active fan speed.					

495	[FC1] Increment / Decrement fan speed	< On / Off	1 bit	-WC---	[1.001] DPT_Switch
With this object, the fan speed can be incremented/decremented when receiving the parametrized 1 bit value. It appears when "**1 bit"** parameter is selected in "Fan manual" -> "Increment/Decrement Fan Speed object"					
495	[FC1] Increment / Decrement fan speed	< 1 byte unsigned	1 byte	-WC---	[5.010] DPT_Counter pulses (0..255)
With this object, the fan speed can be incremented/decremented when receiving the parametrized 1 byte values value. It appears when "**1 byte unsigned"** parameter is selected in "Fan manual" -> "Increment/Decrement Fan Speed object"					
495	[FC1] Increment / Decrement fan speed	< 1 byte signed	1 byte	-WC---	[6.010] DPT_Counter pulses (-128..127)
With this object, the fan speed can be incremented/decremented when receiving the parametrized 1 byte value. It appears when "**1 byte signed"** parameter is selected in "Fan manual" -> "Increment/Decrement Fan Speed object"					
496	[FC1] Purge valve	< 1 = Purge valve / 0 = Nothing	1 bit	-WC---	[1.001] DPT_Switch
With this object, the purge valve cycle parametrized can be activated when receiving the value 1. 0 value will do nothing.					
496	[FC1] Purge valve status	> On / Off	1 bit	R-CT-	[1.001] DPT_Switch
This object will send the purge valve status					
502	[FC1] Heat demand status	> On / Off	1 bit	R-CT-	[1.001] DPT_Switch
This object will send the value 1 in case there is heat demand which will occur when PI > 0%					
503	[FC1] Cool demand status	> On / Off	1 bit	R-CT-	[1.001] DPT_Switch
This object sends the value 1 for a cooling demand (if PI> 0%).					
514 … 52x	Channel switching C1 / C2 - X1 / X2	> On / Off	1 bit	-WC---	[1.001] DPT_Switch
Switching an output channel (number depends on the model variant)					
516 … 52x	Channel status C1 / C2 - X1 / X2	> On / Off	1 bit	R-CT-	[1.001] DPT_Switch
Display of the status of an output channel (number depends on the model variant)					

3 Parameter page: General Settings

Parameter	Settings
DEVICE NAME	**Power Block**
Here a personalized name for each device can be entered. E.g. **Power Block living room**	
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 Power Block 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.	
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 Parameter page: OUTPUTS

Parameter	Settings
CHANNEL A … CHANNEL H	**Binary/Shutter channel** Fan Coil No
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” option. In case “Fan Coil” is selected, 2 channels will be used.	
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: 1 common object = “Central switching/move blind” 2 separate objects = “Central switching” + “Central move”	
Manual control	Param Mode + Test Mode Param Mode Test Mode Disable
The Power Block actuator has 2 push buttons and status LEDs on the front side for each individually channel. 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.	

4.1 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 Power Block Actuator range is also a powerful controller module (logic, timer, counter, etc. module). You can find Advanced Functions: 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
The Power Block actuator has 2 push buttons and status LEDs on the front side for each individually channel. 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.	

4.1.1 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.	

4.1.2 Advanced Functions

<table>
<tr><th>Parameter</th><th>Settings</th></tr>
<tr><td>Central ON/OFF function</td><td><b>No reaction</b>
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</td></tr>
<tr><td colspan="2"><b>No reaction:</b> the channel has no reaction when the Central ON/OFF object/s receive/s a telegram.
<b>Any value = ON:</b> the channel switches ON when the Central ON/OFF object/s receive/s any telegram (no matter whether “0” or “1” is received).
<b>Any value = OFF:</b> the channel switches OFF when the Central ON/OFF object/s receive/s any telegram (no matter whether “0” or “1” is received).
<b>0 = OFF, 1 = ON:</b> the channel switches OFF when the Central ON/OFF object/s receive/s a “0” and switches ON when receiving a “1”.
<b>0 = ON, 1 = OFF:</b> the channel switches ON when the Central ON/OFF object/s receive/s a “0” and switches OFF when receiving a “1”.
<b>Any value = Timer 1 reaction at ON:</b> 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
<b>0 = X, 1 = ON:</b> the channel has no reaction when the Central ON/OFF object/s receive/s a “0” and switches ON when receiving a “1”.
<b>0 = OFF, 1 = X:</b> the channel switches OFF when the Central ON/OFF object/s receive/s a “0” and has no reaction when receiving a “1”.</td></tr>
<tr><td>Additional object</td><td><b>No</b>
Inverted
Toggle only with 0
Toggle only with 1
Toggle with 0 and 1</td></tr>
<tr><td colspan="2"><b>No:</b> this option hides the additional object.
<b>Inverted:</b> 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.
<b>Toggle only with 0:</b> 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”)
<b>Toggle only with 1:</b> 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”)
<b>Toggle with 0 and 1:</b> the output will change its state from OFF to ON or vice versa both when receiving “0” or “1”.</td></tr>
<tr><td>Counters</td><td><b>No</b>
Yes</td></tr>
<tr><td colspan="2">There are two counters (one “Run hour” and one “Switching”) per channel available, both of which can be configured to count up or down.

<b>No:</b> this option hides the counter tab and all its objects and options.
<b>Yes:</b> this option activates the counter tab.</td></tr>
<tr><td>Scenes</td><td><b>No</b>
Yes</td></tr>
<tr><td colspan="2">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.

<b>No:</b> this option hides the Scenes tab and all scene related functions and object for the current channel.
<b>Yes:</b> this option activates the Scene tab, with multiple functions and the Scene object for this channel.</td></tr>
</table>

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 Power Block actuator has 2 push buttons and status LEDs on the front side for each individually channel. 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	

4.1.2.1 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.

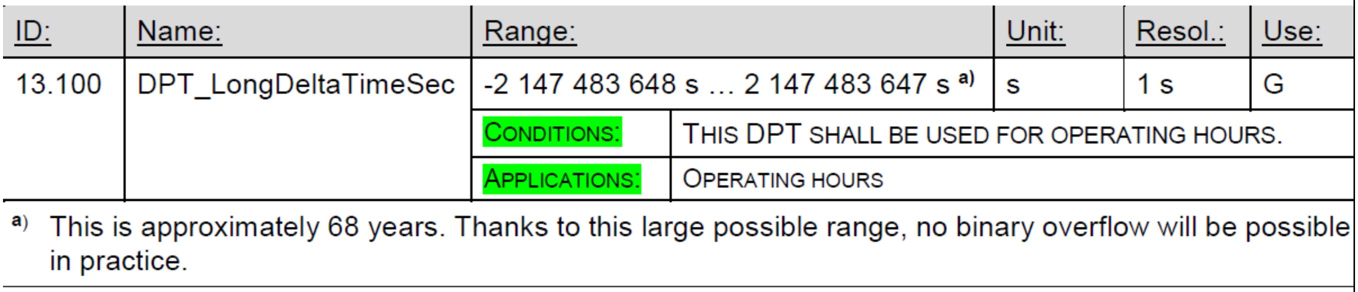
Parameter page: OUTPUTS / Channel A1…X1 (Binary) / ADVANCED FUNCTIONS / Counters / Run hour counter

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.	

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

Parameter	Settings
Data point type of counter	**4 bytes**

Usually, a Run hour counter has a 4 bytes value, counting in seconds, according DTP 13.100.



ID:	Name:	Range:	Unit:	Resol.:	Use:
13.100	DPT_LongDeltaTimeSec	-2 147 483 648 s … 2 147 483 647 s [a]	s	1 s	G
		CONDITIONS:	THIS DPT SHALL BE USED FOR OPERATING HOURS.		
		APPLICATIONS:	OPERATING HOURS		

[a] This is approximately 68 years. Thanks to this large possible range, no binary overflow will be possible in practice.

Parameter	Settings
Initial value run hour counter *Attention! After programming this value will only be overwritten if the new starting value is changed.*	**No** Yes
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. Take into account that the additional counter 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 ***Attention! 0 = Deactivated***	0

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) 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:** 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 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.	

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. Take into account that the additional counter 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.	

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.	

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.	

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. Take into account that the additional counter 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) Attention! Both counter & alarm objects will be set to zero	**Reset to 0 and start again** Stay at maximum
<u>Important note</u>: 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.	

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

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. However, 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. Take into account that the additional counter 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.	

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
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.	

4.1.2.2 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.	

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) **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)	

4.1.2.3 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:

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	

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.	

- Factor changeable by object / Remaining time cyclic sending	**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 Additionally, to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value until the timer finish. In order to disable this function, the “T” flag must be deactivated.	
Advanced staircase function	**No** Yes
Here the advanced functions can be activated.	

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 ON telegrams received. This resulting time will never exceed the parameterized maximum staircase in the option “Maximum staircase time Base/Factor” It is important to keep in mind that the multiplication will only be done starting from the first triggering telegram (so, the Multiplying staircase function will only be executed when starting the staircase, not during execution). Therefore, these ON telegrams may not be longer than 1 second apart. 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). 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). But this function will only be executed after more than 1 second has elapsed between the triggering events of the timer (if less than 1 second, see behaviour in section MULTIPLY STAIRCASE).

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.

The channel, 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.

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.	
- Factor changeable by object / Remaining time cyclic sending	**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”. Additionally, to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value until the timer finish. In order to disable this function, the “T” flag must be deactivated.	
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.A. Infinite repetitions can be achieved by using the value 65535. In order to deactivate the blinking, just enter the value 0.	

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.	

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

Parameter	Settings
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.	

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.	

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.	

4.1.2.4 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	**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
Type of object for deactivation	
- Value	0 **1**
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.

4.1.2.5 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).

4.2 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 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	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.	

4.2.1 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: 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. STOP: So, because this time is shorter, the blind will stop before the time for long operation in the push button has elapsed. 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 Power Block Actuator range is also a powerful controller module (logic, timer, counter, etc. module). You can find Advanced Functions: 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 Power Block actuator has 2 push buttons and status LEDs on the front side for each individually channel. 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.	

4.2.2 Bus failure

<table>
<tr><th>Parameter</th><th>Settings</th></tr>
<tr><td>Reaction on bus voltage failure</td><td>Unchanged
Up
Down
Stop</td></tr>
<tr><td colspan="2">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).</td></tr>
<tr><td colspan="2">**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.</td></tr>
<tr><td>Reaction on bus voltage recovery</td><td>**Stop**
Up
Down
Position
Move to slat and blind position
Preset
Recovery status before bus failure</td></tr>
<tr><td colspan="2">**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.</td></tr>
<tr><td colspan="2">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</td></tr>
</table>

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.

4.2.3 Advanced functions

Parameter	Settings
Precision time	**No** Yes
The advantage of the precision time function is that now it is possible to: Different travel time for movement down Control and positioning the slits of the shutter Positioning the shutter/blind in the true percentage height, obtaining a real shutter positioning for the end-customer using the correction curve **No:** this option hides the Precision time tab. **Yes:** this option activates the Precision time tab, with the following functions and objects for this channel.	
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. 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”): 1 common object = “Central switching/move blind” 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.

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 travel time from upper to lower position has been defined, this is taken into account.	
Manual control	No **Yes**
Attention! Manual control must be activated in outputs	
The Power Block actuator has 2 push buttons and status LEDs on the front side for each individually channel. 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.	

4.2.3.1 Precision time

Different travel time for movement DOWN

Parameter	Settings
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.	

Parameter page: General settings/OUTPUTS / Channel X1 (slat/blind) / Extended functions / accuracy Time/slot Function

Parameter	Settings
Slit function	**No** Yes
This function is especially interesting when the height of the shutters is too great, allowing to the end-user to control the amount of slits open in order to bring natural light into the building. When the Slit positioning object receives a percentage value, the shutter will be moved until the bottom is touching the frame of the window, e.g. To close the shutter with all the slits open: Slit object must be set to the value 0%. The status objects would therefore stay as follows: - Slit status position = 0% - Shutter status position = 100% To close the shutter with all the slits closed: Silt object must be set to the value 100% (it is the same than if the shutter positioning object receives a value = 100%.) The status objects would therefore stay as follows: - Slit status position = 100% - Shutter status position = 100%	

Slit time base Slit time factor	**100 ms** 40
This is the travelled time since the bottom of the shutter starts to touch the window frame with all the slits open, until all the slits are completely closed (shutter 100% closed).	

Shutter position correction curve

Parameter	Settings
Shutter position correction curve	**No** Yes
It is very typical to send a value for positioning the shutter, i.e. 50%, and when it finishes the movement, the true and visible position reached is the 70%. To solve the above problem, this function corrects the usual non-linear up/down rolling error in order to achieve the true shutter position.	
Time from 0% to 50% Factor	**100 ms** 80
For the measurement of this time, the shutter must be moved to the top position in order to reach the 0% value. Then, the time considered must be from the top till the true 50% position. This time is needed to correct the non-linear up/down rolling error.	

More precision for Up movement

Parameter	Settings
More precision for Up movement	**No** Yes
The function “Shutter position correction curve” fixes the error produced in most cases. In some cases, due to the excessive weighting of the shutter, more precision time is required. This parameter offers the possibility to give more accuracy in the positioning when the “Shutter position correction curve” parameter is not enough.	
Time from 100% to 50% Factor	**100 ms** 120
For the measurement of this time, the shutter must be moved to the bottom position in order to reach the 100% value. Then, the time considered must be from the bottom till the true 50% position. Using this time, more precision is given to correct the non-linear up/down rolling error.	

4.2.3.2 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.	

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.	

4.2.3.3 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 objects to save the current movement and slat position as the new preset value by sending a 1 to this object.	

4.2.3.4 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:

<table>
<tr><th>Parameter</th><th>Settings</th></tr>
<tr><td>Alarm 1
…
Alarm 8</td><td>Nothing
Block channel as is
Move Up
Move Down.
Move to position
Move to preset</td></tr>
<tr><td colspan="2">**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.</td></tr>
<tr><td>Behaviour at end of all alarms</td><td>Nothing
Move Up
Move Down
Move to position
Move to preset
Set to tracked state</td></tr>
<tr><td colspan="2">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.
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.</td></tr>
</table>

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.*

4.2.3.5 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	

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. In addition, the “Behaviour at enabling” will only be executed with no active & acknowledged channel alarms.*

4.2.4 Status shutter

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.	
Send 1 byte slit position status telegram	**No** Yes

If “Yes” is selected on this menu, the “Status slit position” object will be activated. Its value will be updated as follow:

When the “Slit positioning” object receives a percentage value, the shutter will be moved until the bottom is touching the frame of the window, e.g.

To close the shutter with all the slits open:
Slit object must be set to the value 0%.

The status objects would therefore stay as follows:
- Slit status position = 0%
- Shutter status position = 100%

To close the shutter with all the slits closed:
Slit object must be set to the value 100%
(It is the same than if the shutter positioning object receives a value = 100%.)

The status objects would therefore stay as follows:
- Slit status position = 100%
- Shutter status position = 100%

5 Parameter page: FAN COIL

5.1 Fan Coil Settings

Parameter	Settings
Type of Fan Coil	**Heat/Cool (2 pipes)** Heat (2 pipes) Cool (2 pipes) Heat/Cool (4 pipes)
Heat (2 pipes): For fan coil systems with only hot air **Cool (2 pipes**): For fan coil systems with only cold air **Heat/Cool (2 pipes):** For fan coil systems with both hot and cold air in 2-pipes water facilities. **Heat/Cool (4 pipes):** For fan coil systems with both hot and cold air. Toggle between hot and cold air is supported by independent fan coil units in 4-pipes water facilities.	
Delay between Heat/Cool mode changes	**No** Yes
A delay may be applied when a change between Heating and Cooling occurs. This option is available when setting **Heat/Cool (2 pipes) or Heat/Cool (4 pipes)** as the Fan coil type.	
Base Factor	**1 sec.** 1
When this option is active, the default 1 sec. delay is visible. This option allows the configuration of the time needed by those HVAC devices which need an additional time to switch between Heating and Cooling (or vice versa), to vary their behaviour.	
ON/OFF object	**No** Yes
Each Power Block fan coil controller supports enabling the ON/OFF object to fully activate and deactivate the fan coil system. This can be very useful to link with the appropriate thermostat when the latter has the same control object. This allows an easy way to switch the fan coil ON/OFF.	
Disable manual buttons on device	**No** All Individually
No: Manual control of the 3 fan speeds and valve with the push buttons on the device is supported. **All**: Manual control is fully disabled both in the fan and the valve. **Individually**: Manual control for any of the 3 fan speeds and the valve can be blocked individually. By activating this option, the tab "*Manual device buttons"* shows up with the allowed parameters.	
Behaviour at bus failure/recovery	**No** Yes
The behaviour of the different fan coil functionalities on bus recovery can be defined here. By activating this option, the tab "*Behaviour at bus failure/recovery"* shows up with the allowed parameters.	
Operation modes (Fan & Valve)	**1 operation mode** 2 operation modes 3 operation modes 4 operation modes

Operation modes help us define preset behaviours in the fan coil, applying restrictions to both the fan and the valve. By default, operation modes are preset with the following sample parameters (that can be adapted to the needs of each installation): Within the tab Fan Speed we can find further tabs to restrict or allow the fan options for each mode: **1 operation mode**: - **Fan OFF, manual:** In Manual Mode, the fan speed might not be set to OFF. **2 operation modes**: - **Max:** In Auto Mode, the fan speed might only be set to Fan 3 and OFF. **3 operation modes**: - **Eco:** In Auto Mode, the fan speed might only be set to Fan 1 and OFF. **4 operation modes: - User:** In Manual Mode, the fan speed might only be set to Fan 1. Within the tab Valve we find the tab "*Operation mode"*; here we can restrict or allow the valve's positioning values for each mode. In this case, only one tab is enabled to configure all 4 operation modes in the valve; there are sample values for the above mentioned modes. By activating any of these options, the relevant tabs for each one are shown in the following tabs: "*Fan Speed" -> Operation mode 1..4".* and "*Valve -> Operation modes"*	
Behaviour when exiting operation mode	**Set to tracked state**
The behaviour of the fan when exiting any of the enabled modes is defined here. The fan speed and the valve will be positioned according to the current object values and parameters when exiting the active mode.	
Advanced functions	**No** Yes
The following advanced functions can be activated here	
Scenes & Day/Night object	**No** Yes
The scenes functionality, as well as the Day/Night object can be enabled here. We might define the behaviour	
Alarm function	**No** Yes
Two alarm tabs are enabled: "*Fan Speed -> Alarms fan*" and another one in "*Valve -> Alarms valve*"	
Thermostat monitoring	**No** Yes
The Thermostat monitoring functionality is activated within the Fan Speed and Valve tabs, as well as the following parameters:	
Thermostat monitoring time Factor	**1 min** **10**
The monitoring time for thermostat can be set here. Within this time of at least one PI value from the thermostat must be received; otherwise, an error will occur (in which case the fan and valve behaviour can be defined via parameters).	
Switch FC OFF with thermostat error	**Error = Stay ON** (Set Fan & Valve in own tabs) Error = Switch FC OFF / Set to tracked state
Error = Stay ON (Set Fan & Valve in own tabs): The fan and valve behaviour can be defined here when an error is detected. The behaviour parameters can be set in the *"Fan Speed" and "Valve" tabs.* Error = Switch FC OFF / Set to tracked state: The fan coil is switched off when an error occurs. When the error stops, the fan coil stays in the status that was actually due, as if the error had never happened.	

5.1.1 Manual device buttons

Parameter	Settings
Fan speed 1 (Output 1)	**Enable** Always disable
Manual control of the fan speed 1 can be enabled/disabled individually.	
Fan speed 2 (Output 2)	**Enable** Always disable
Manual control of the fan speed 2 can be enabled/disabled individually.	
Fan speed 3 (Output 3)	**Enable** Always disable
Manual control of the fan speed 3 can be enabled/disabled individually.	
Heating/Cooling valve (Output 4)	**Enable** Always disable
Manual control of the control valve can be enabled/disabled individually.	

5.1.2 Behaviour at bus failure/recovery

Parameter	Settings
HEAT/COOL MODE Behaviour at bus recovery	**Unchanged** Read request Heat mode Cool mode
Unchanged: The mode that was enable previous to the bus failure (heat/cool) stays active on bus recovery. **Read request**: On bus recovery the communication object sends a read request to the bus to set the operation mode heat/cool. Note: Attention!! With no answer after read request, the mode will be the one existing before the bus failure. **Heat mode:** On bus recovery, the Heat mode is set. **Cool mode**: On bus recovery, the Cool mode is set.	
Send status value	**No** Yes
On bus recovery, the object value is sent after the delay configured in the "General Settings" tab.	
FAN SPEED (AUTO/MANUAL) Behaviour at bus failure	**Unchanged** Manual Fan OFF Manual Fan 1 Manual Fan 2 Manual Fan 3
Unchanged: On bus failure, the current speed stays active; in other words, the relays relevant for the speed control stay interlocked in their position (open/closed). **Manual Fan OFF:** All fan outputs are switched off, the fan being fully shutdown. **Manual Fan 1:** The fan speed 1 stays enabled. **Manual Fan 2**: The fan speed 2 stays enabled. **Manual Fan 3:** The fan speed 3 stays enabled.	
Behaviour at bus recovery	**Unchanged** Read request Manual Fan OFF Manual Fan 1 Manual Fan 2 Manual Fan 3 Fan auto Recovery status before bus failure Manual fan last speed

Unchanged: On bus recovery the speed configured last stays active. **Read request:** On bus recovery the communication object sends a read request to the bus to set the fan speed. Note: **Attention!** With no answer after read request, the mode will be the one existing before the bus failure. **Manual Fan OFF:** All fan outputs are switched off, the fan being fully shutdown. **Manual Fan 1:** The fan speed 1 is enabled. **Manual Fan 2**: The fan speed 2 is enabled. **Manual Fan 3:** The fan speed 3 is enabled. **Fan auto:** The automatic mode of fan speed control is activated. **Recovery status before bus failure:** The fan status active previous to the bus failure is recovered. **Manual fan last speed:** The last speed previous to the bus failure is set, but not in manual mode.	
Send status value	**No** Yes
On bus recovery, the object value is sent after the delay configured in the “General Settings” tab.	
THERMOSTAT MONITORING: FAN BEHAVIOUR	
Behaviour at bus recovery	**Unchanged**
In case the Thermostat monitoring error was active, the fan speed will remain unchanged on bus recovery.	
OPERATION MODE	
Behaviour at bus recovery	**Unchanged** Exit operation modes
Unchanged: On bus recovery the mode configured last stays active. **Exit operation modes:** Any operation mode that might have been active previous to the bus failure will be exited.	
Send status value	**No** Yes
On bus recovery, the object value is sent after the delay configured in the “General Settings” tab.	
VALVE POSITION	
Behaviour at bus failure	**Unchanged** On Off
Unchanged: On bus failure the status of the valve's last position stays active. **On:** Sets the valve's to ON **Off:** Sets the valve's to OFF	
Behaviour at bus recovery	**Set to tracked state**
Set to tracked state: The valve's relay is set to the corresponding actual estimated status (with the last PI value of the thermostat as received previous to bus failure).	
THERMOSTAT MONITORING: VALVE BEHAVIOUR	
Behaviour at bus recovery	**Unchanged**
In case the Thermostat monitoring error was active, the position of the valve will remain unchanged on bus recovery.	

5.1.3 Special operation mode 1(Tab fan speed)

Description based on Special operation mode 1 (-Deny Fan OFF manual). 3 further special operating modes are available to the user (Max, Eco and User). The presettings can be adapted by the user to the current requirements.

Parameter	Settings
Description	**- Deny Fan OFF manual**
Descriptive name of the Operation Mode 1..4	
Apply operation mode for fan when in	**Manual mode** Auto mode Both
Manual mode: The operation mode will only be applied when the fan mode is set to “Manual” **Auto mode:** The operation mode will only be applied when the fan mode is set to “Auto” **Both**: The operation mode will be applied when the fan mode is set to both “Manual and Auto mode” All operation mode settings for the fan (i.e. restrictions, etc.) will only be applied to the fan when the above fan mode selection is active. With these default settings, “Manual mode”. When the Fan is in Auto mode, the fan will work as if the operation mode is not active; but when the fan mode is changed to Manual mode, then this Operation mode will be applied to the fan. If “Auto”/”Manual” modes are not selected, the system will not apply the operation mode. For example, with “Deny Fan OFF manual”, the Fan OFF action is only restricted during the manual mode; nevertheless, it would be allowed if the user selects the “Auto” mode of the fan. Attention! There are no priorities, the last operation mode received will be active.	
Operation mode trigger value	**ON -> Activated, OFF -> Exit** OFF -> Activated, ON -> Exit
ON -> Activated, OFF -> Exit: The mode is activated with value 1 and deactivated with value 0. **OFF -> Activated, On -> Exit:** The mode is activated with value 0 and deactivated with value 1. Each of the 4 operation modes has a 1-bit trigger object that can be individually configured as explained above.	
Restrict to actual fan speed	**No** Yes
No: Define the speeds that can be active during the activation of the mode. **Yes**: The fan speed will be restricted to the one operating in that given moment in time; no other speed will be allowed while the mode is active. This can be useful as a do-not-disturb function. In the “Deny Fan OFF manual” example, the configuration of the following parameters (Allow Fan OFF, Allow Fan speed 1, Allow Fan speed 2 y Allow Fan speed 3) show how avoid to switch the FAN OFF manually.	
Allow Fan OFF	**No** Yes
No: During the activation of the mode, the FAN OFF operation will be restricted. **Yes**: During the activation of the mode, the FAN OFF operation will be allowed.	
Allow Fan speed 1	**Yes** No
No: During the activation of the mode, the Fan speed 1 operation will be restricted. **Yes**: During the activation of the mode, the Fan speed 1 operation will be allowed.	
Allow Fan speed 2	**Yes** No
No: During the activation of the mode, the Fan speed 2 operation will be restricted. **Yes**: During the activation of the mode, the Fan speed 2 operation will be allowed.	
Allow Fan speed 3	**Yes** No
No: During the activation of the mode, the Fan speed 3 operation will be restricted. **Yes**: During the activation of the mode, the Fan speed 3 operation will be allowed.	

Attempting to change to a restricted fan speed causes	**Change to next higher fan speed** Change to next lower fan speed No change
Select the action that shall be executed if there was an attempt to set a not-allowed speed while the mode is active. **Change to next higher fan speed:** Switch to the next highest speed allowed **Change to next lower fan speed:** Switch to the next lowest speed allowed **No change**: Keep current speed and make no changes	
Behaviour after ETS download	**Enabled** Disabled Unchanged
The behaviour of the operation mode after downloading the application program from the ETS is defined here. **Enabled:** The operation mode is enabled. **Disabled:** The operation mode is disabled. **Unchanged:** No action is performed; the mode stays as it was previous to the ETS download. In order to avoid conflicts between the different modes, this parameter is only available in “Operation mode 1”.	
Temporized operation mode, return to normal after	**No** Yes
Once the operation mode has been activated, it will automatically exit the operation mode after the time established in the following parameters has elapsed: **Base: 1h** **Factor 1**	

5.1.4 Operation mode (Valve tab)

Description based on special operating mode 1, valve position (manual operation). Three further special operating modes (valve position) are available to the user. The presettings can be adapted by the user to current operating requirements.

Parameter	Settings
Operation mode 1..4 valve position Apply operation mode for valve when in	 **Manual mode** Auto mode Both
Manual mode **Auto mode:** **Both:** All operation mode settings for the valve (i.e. restrictions, etc.) will only be applied to the valve when the above fan mode selection is active. With these default settings, “Manual mode”. When the Fan is in Auto mode, the valve will work as if the operation mode is not active; but when the fan mode have changed to Manual mode, then this Operation mode will be applied to the valve. If “Auto”/”Manual” modes are not selected, the system will not apply the operation mode. Taking for example “Deny Fan OFF manual”, the valve is allowed to go all the way through from 0 to 100% during the manual model, where “Operation mode 1” is applied. *Attention!* There are no priorities; the last operation mode received will be active.	
Allow closing the valve with PI = 0%	**Yes** No
When the mode is active, it either allows or prevents the valve from closing in PI value = 0%.	
Permitted valve stroke	**Allow valve from 0%** Allow valve to 100%
Here we can define the valve range when the operation mode is active: **Allow valve from 0%:** Initial permitted value for the positioning of the valve Allow valve to 100%: Final permitted value for the positioning of the valve	
Heating/Cooling valve (Output 4)	**Enable** Always disable
Manual control of the control valve can be enabled/disabled individually.	

5.1.5 Scenes 1..4

Description based on scene 1. Three additional scenes are available to the user. The presettings can be adapted by the user to current operating requirements.

Parameter	Settings
Scenes	**No** Yes
The parameters relevant to scenes 1..4 are shown/hidden. Up to 4 scenes and 3 events in each scene can be configured to establish different fan speeds in each one of them.	
Scene name	**Descriptive name for the scene**
Scene number	**Scene 1 .. Scene 64**
Select here the number of the scene which will trigger the scene events sent to the bus.	
1 bit scene objects	**No** Yes
No: The 1-bit object is hidden **Yes**: The 1-bit object is shown The 1-bit object can be individually activated or deactivated to launch the scene.	
Possible to save scene	**No** Yes
Select here if the values to be sent by the event objects will be updated by the new ones received from the bus in these objects when the scene is saved.	
Event 1..3 Fan Speed	 **Nothing** Manual Fan speed 1 Manual Fan speed 2 Manual Fan speed 3 Manual Fan Off Fan Auto
Select here the fan speed and Auto/Manual mode which should be set when the scene is triggered.	
Delay	**No** Yes
Enable here a delay between the current event and the next one, which only starts running after completion of the previous event. **Delay base:** 1s **Factor:** 1	

Day/Night object	ON = Day / OFF =Night
	OFF = Day / ON =Night

Configure here the type of value to execute the scene linked to the Day or Night mode. **ON = Day / OFF =Night:** Enable the Day scene on reception of value ON. Enable the Night scene on reception of value OFF. **OFF = Day / ON =Night:** Enable the Day scene on reception of value OFF. Enable the Night scene on reception of value ON.	
Reaction on day	No reaction Play scene 1 Play scene 2 Play scene 3 Play scene 4
Select the scene to be launched when the Day mode is activated in the Day/Night object.	
Reaction on night	No reaction Play scene 1 Play scene 2 Play scene 3 Play scene 4
Select the scene to be launched when the Night mode is activated in the Day/Night object.	

5.1.6 Alarms fan (Fan tab)

Parameter	Settings
Forced fan speed on alarm 1..8	**Nothing** Force actual Manual Fan Off Manual Fan speed 1 Manual Fan speed 2 Manual Fan speed 3 Fan Auto
Decide here the behaviour of the fan when enabling each one of the 8 alarms already existing in the “*General settings -> Advanced functions-> Alarms”*. The following options are available as long as the selected alarm is active: **Nothing:** No action takes place **Force actual:** The speed currently active is forced. **Manual Fan Off:** The fan switch off or speed 0 in Manual mode are forced. **Manual Fan speed 1:** Fan speed 1 is forced in Manual mode. **Manual Fan speed 2:** Fan speed 2 is forced in Manual mode. **Manual Fan speed 3:** Fan speed 3 is forced in Manual mode. **Fan Auto:** Auto mode is forced Attention!! Priorities: Alarm 1 (highest)...8 (lowest)	
Unforced fan speed at end of all alarms	**Keep actual** Manual Fan Off Manual Fan speed 1 Manual Fan speed 2 Manual Fan speed 3 Fan Auto Set to tracked state
Decide here the behaviour of the fan on completion of all the alarms that had been active. **Keep actual:** The speed currently active is kept. **Manual Fan Off:** The fan is switched off or speed 0 set in Manual mode. **Manual Fan speed 1:** Fan speed 1 is set in Manual mode. **Manual Fan speed 2:** Fan speed 2 is set in Manual mode. **Manual Fan speed 3:** Fan speed 3 is set in Manual mode. **Fan Auto:** Auto mode is set **Set to tracked state:** The speed of the fan is set to match the speed that it should have had if no alarm had been triggered.	

5.2 Fan speed

<table>
<tr><th>Parameter</th><th>Settings</th></tr>
<tr><td>Type of Fan switching</td><td><b>Single (Only 1 ON at time)</b>
Multiple (Switch outputs sequentially ON)</td></tr>
<tr><td colspan="2">The type of fan used in the fan coil is defined here; this option determines the behaviour of the actuator outputs for the electric control of the fan:

<b>Single (Only 1 ON at time):</b> Only one output is activated at a time:
- Fan speed 0: No output is activated
- Fan speed 1: Only output 1 of the fan is activated
- Fan speed 2: Only output 2 of the fan is activated
- Fan speed 3: Only output 3 of the fan is activated

<b>Multiple (Switch outputs sequentially ON)</b>
- Fan speed 0: No output is active
- Fan speed 1: Output 1 of the fan is activated
- Fan speed 2: Outputs 1 and 2 of the fan are activated
- Fan speed 3: Outputs 1, 2 and 3 of the fan are activated

Important note: Previous to the commissioning of the fan coil actuator, it’s important to identify the type of control required for the control of the 3 speeds. In case of a wrong interpretation, irreparable electrical damages can be caused to the fan of the fan coil system.</td></tr>
<tr><td>Delay between switchings

Factor</td><td><b>500ms</b>

<b>1</b></td></tr>
<tr><td colspan="2">This option is active when parameter “Single (Only 1 ON at time)” has been selected.

The waiting time in which all the fan outputs are OFF before activating the relevant output for the new speed can be defined here.</td></tr>
<tr><td>Number of fan speeds</td><td><b>3</b>
2
1</td></tr>
<tr><td colspan="2">The number of speeds allowed by the fan coil system are set here.</td></tr>
<tr><td>Remaining time to change filter</td><td><b>No</b>
Yes</td></tr>
<tr><td colspan="2">You can enable the <i>“Fan speed -> Filter remaining time”</i> tab here; this tab shows the parameters necessary to notify when the air filters of the fan coil system need replacing. This is in other words a backwards counter that only decrements the remaining time while the fan is ON.</td></tr>
<tr><td>Fan speed timers/delay/cyclic</td><td><b>No</b>
Yes</td></tr>
<tr><td colspan="2"><b>No:</b> Parameters are hidden
<b>Yes</b>: It shows multiple timer options for configuration in different scenarios.</td></tr>
<tr><td>Temporized forced initial fan speed. When FC switches ON</td><td><b>No</b>
Yes</td></tr>
<tr><td colspan="2">The fan is forced into a specific speed when the communication object “FC ON/OFF” receives the value ON.
<b>No</b>: Parameters are hidden
<b>Yes</b>: The following parameters are shown</td></tr>
</table>

Temporized forced initial speed	**Speed 1** Speed 2 Speed 3
Forced speed when the fan coil switches ON	
Allow manual speed changes in initial force speed	**Yes** No
Switching speed manually is allowed during the forced time period.	
Duration for forced fan speed	**1 min** **10**
Duration of the forced speed time on fan coil activation	
Fan delay when FC switches ON (warm/cool start)	**No** Yes
A delay in activating the fan is allowed when the object “FC ON/OFF” receives the value ON. Thus, the air supply at room temperature is avoided when hot/cold water is still not available in the pipes to supply air at the correct temperature. It can be very useful in water circuits where there is a relevant distance between the fan coil unit and the water production system. Attention! Delay only starts after first valve demand when FC switches ON **No:** Parameters are hidden **Yes:** The following parameters are shown:	
Starting delay (Ignores Fan ON delay) Factor	**1 min** **5**
The initial delay in this example is 5 minutes. 5 minutes after having switched on the fan coil unit, the fan will start; in the meantime, it will remain disabled. It is important to highlight here that, while this timing is ON, the timing of the Fan ON delay function is ignored.	
Delay fan	**No** Only with Fan auto Only with Fan manual Both
A timer is set for the fan, which will start when one of the following changes takes place: - From any speed to Fan OFF - From Fan OFF to any speed The mode Auto/Manual where it should apply can also be defined: **No:** No timer **Only with Fan auto:** It applies only in Auto mode **Only with Fan manual:** It applies only in Manual mode **Both:** It applies both in Auto/Manual mode The following parameters are enabled whenever one of the 3 timers has been selected:	
Fan delay Base Factor	 **1 min** 1

Additional cyclic ventilation	**No** Yes, always (Even when FC is OFF) Yes, only in Auto mode Yes, only in Manual mode Yes, Auto & Manual mode Yes, only when FC is OFF
The air recirculation in one or more rooms, when necessary, can set here; both the speed and the activation frequency can be configured. The available options are: **No:** Hidden parameters **Yes, always (Even when FC is OFF):** the additional cyclic ventilation will be activated automatically after programming the device or connecting it to the system, independent whether the fan coil is ON or OFF. **Yes, only in Auto mode**: the additional cyclic ventilation will only be activated when the fan coil switches to Auto mode. **Yes, only in Manual mode**: the additional cyclic ventilation will only be activated when the fan coil switches to Manual mode. **Yes, Auto & Manual mode:** the additional cyclic ventilation will only be activated both with Auto and Manual mode **Yes, only when FC is OFF:** the additional cyclic ventilation will only be activated when the fan coil is switched OFF (making use of the communication object FC On/Off) Attention! Priorities: Alarms -> Operation modes -> Additional cyclic -> Normal operation	
Minimum Fan Speed at cyclic ventilation	**Speed 1** Speed 2 Speed 3
Minimum speed to activate the cyclic ventilation	
Cyclic Fan switching: Switch Fan ON every Factor	**1h** **5**
Activation frequency. In this example, it will be activated every 5 hours	
Fan ON duration	**1 min** **60**
Duration of ventilation on each activation. In this example, the duration is 60 minutes every 5 hours.	
Thermostat monitoring: Fan behaviour	**No** Yes
It shows the parameters to establish the fan operation when the thermostat monitoring function causes an error.	
Thermostat monitoring: Fan behaviour	**Error = Switch fan OFF**

5.3 Fan Auto

Parameter	Settings
The following parameters are available to achieve an automatic control of the fan speed	
Type of control signal	**PI (0..100%)** Temperature difference
There are two different types of input control: **PI (0..100%):** Value input by 1-byte PI (proportional integral) scaling object **Temperature difference:** Value inputs using the room temperature and the setpoint temperature.	
PI (0..100%) (if this type of input control is activated) The fan speed is established taking into account the values received from the PI. Is the value lower, the speed decreases (less difference between the room and the setpoint temperature). Is the value higher, the speed increases (bigger difference between the room and setpoint temperatures)	
Fan OFF	**Yes, If PI value is lower/equal “Speed I -Hyst.”** No
Speed 0 can be enabled or restricted in the Auto mode. **Yes, If PI value is lower/equal “Speed I -Hyst.”:** The Fan OFF speed can be enabled when the PI value is lower or equal to the value established as threshold for speed 1 minus the hysteresis value. **No:** Speed 0 is not allowed in the Auto mode. Taking into consideration the default values as an example, it looks like this:	
Speed 1 from Hysteresis	**1** **1**
If speed 1 is active: Switch to speed 2: -> When the PI value received is equal/higher than the threshold value (40) Switch to speed 0 -> When the PI value received is lower than the threshold value (1) – Hysteresis (1); that is, 0.	
Speed 2 from Hysteresis	**40** **5**
If speed 2 is active: - Switch to speed 3: -> When the PI value received is equal/higher than the threshold value (70) - Switch to speed 1: -> When the PI value received is lower than the threshold value (40) – Hysteresis (5); that is, 35.	
Speed 3 from Hysteresis	**70** **5**
If speed 3 is active: - Switch to speed 2 -> When the PI value received is lower than the threshold value (70) – Hysteresis (5); that is, 65. *Attention!* To set or increase a Speed: Value received >= “Speed X from” To decrease a Speed: Value received <= “Speed X from” – “Hyst”	

Temperature difference (if this type of input control is activated)

The fan speed is established taking into account the values received from the room and the setpoint temperature. The larger the difference between them both, the higher the speed. The smaller the difference between them both, the slower the speed.

There are 2 objects available for the value input of both reference temperatures.

Attention: Temperature difference between actual and setpoint temperature.

Fan OFF	**Yes, If Temp Diff is lower "Speed I -Hyst."** No

Speed 0 can be enabled or restricted in the Auto mode.

Yes, If Temp difference is lower "Speed I -Hyst.": The Fan OFF speed can be enabled when the temperature difference is lower than the value established as threshold for speed 1 minus the hysteresis value.
No: Speed 0 is not allowed in the Auto mode.

Taking into consideration the default values as an example, it looks like this:

Speed 1 from Hysteresis	**0** **0.5**

If speed 1 is active:

- Switch to speed 2: -> When the temperature difference is equal/higher than the threshold value (3)
- Switch to speed 0 -> When the temperature difference is lower than the threshold value (0) – Hysteresis (0.5); that is, -0.5.
-

Speed 2 from Hysteresis	**3** **0.5**

If speed 2 is active:

- Switch to speed 3: -> When the temperature difference is equal/higher than the threshold value (5)
- Switch to speed 1: -> When the temperature difference is lower than the threshold value (3) – Hysteresis (0.5); that is, 2.5.
-

Speed 3 from Hysteresis	**5** **0.5**

If speed 3 is active:

- Switch to speed 2 -> When the temperature difference is lower than the threshold value (5) – Hysteresis (5); that is, 4.5.

Attention!!
To set or increase a Speed: Value received >= "Speed X from"
To decrease a Speed: Value received <= "Speed X from" – "Hyst"

Switch Fan OFF when valve is closed	**No** Yes

The Fan OFF speed can be set when the valve stays closed during the appropriate period within the PWM cycle derived from the PI value.

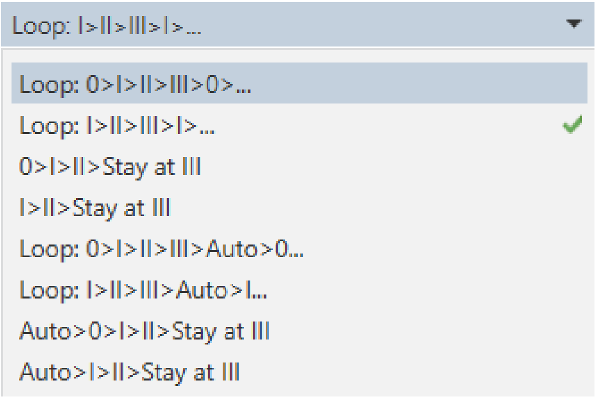
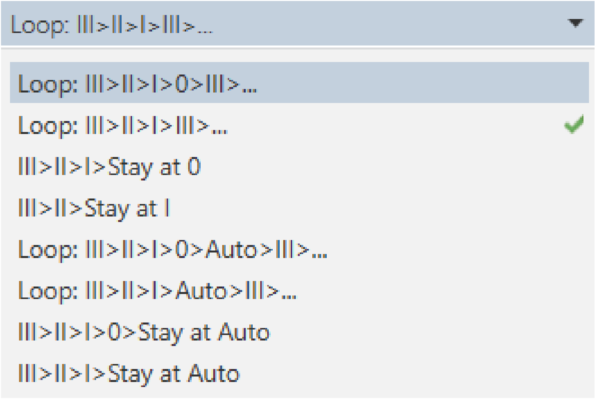
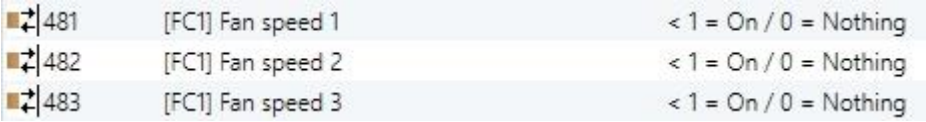
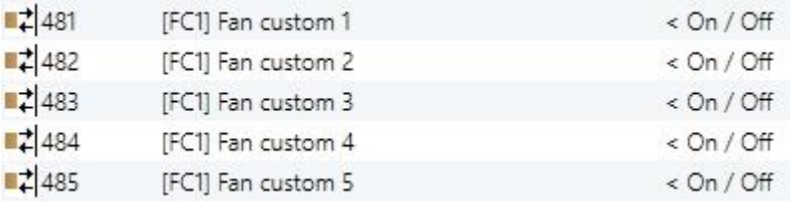
Min. maintaining time in fan speed	

In the Auto mode, the set speed will remain the same for a minimum time before switching to another speed. Configure the minimum time here:

- **Base: 1 min**
- **Factor: 5**

5.4 Fan Manual

Parameter	Settings
The following parameters are available to achieve a manual control of the fan speed	
Manual fan speed 1 byte object	**No** Scaling 0..100% Unsigned 0..255 value Both
Control by standard objects 1 byte scaling & 1 byte unsigned The following standardized objects support the manual speed control in two different ways: **No:** The manual control objects are hidden **Scaling 0..100%:** The 1 byte percentage control object is shown The standardized values ranges for the speed control are as follows: - Fan speed 0 = 0% - Fan speed 1 = 0.4 - 33,3% - Fan speed 2 = 33.7 – 66.7% - Fan speed 3 = 67.1 - 100% **Unsigned 0..255 value**: The 1 byte unsigned control object is shown **Both**: Both the 1 byte unsigned and 1 byte percentage control objects are shown - Fan speed 0 = 0 - Fan speed 1 = 1 - Fan speed 2 = 2 - Fan speed 3 = 3	
Increment/Decrement Fan speed object	**1 bit** **1 byte unsigned** **1 byte signed**
Control via Increment/Decrement objects Additionally to the standardized 1 byte control objects, the device supports control via the following objects (establishing values for the increase or decrease of the speeds and having them sent repetitively): **1 bit** **1 byte unsigned** **1 byte signed** In all 3 cases, the speed increase and decrease value can be set, thus adapting the value to the corresponding DPT. The following parameters are available for this function:	
Value to increment	**1**
Value to decrement	**0**

Increment sequence	 Loop: I>II>III>I>... Loop: 0>I>II>III>0>... Loop: I>II>III>I>... ✓ 0>I>II>Stay at III I>II>Stay at III Loop: 0>I>II>III>Auto>0... Loop: I>II>III>Auto>I... Auto>0>I>II>Stay at III Auto>I>II>Stay at III
The allowed sequences for the fan speed are shown when sending the increase value	
Decrement sequence	 Loop: III>II>I>III>... Loop: III>II>I>0>III>... Loop: III>II>I>III>... ✓ III>II>I>Stay at 0 III>II>Stay at I Loop: III>II>I>0>Auto>III>... Loop: III>II>I>Auto>III>... III>II>I>0>Stay at Auto III>II>I>Stay at Auto
The allowed sequences for the fan speed are shown when sending the decrease value	
Accept Increment/decrement changes only after (x100ms)	**8**
Manual fan speed 1 bit object	**No** Yes, 3 x 1 bit Yes, custom
No: Parameters are hidden **Yes, 3 x 1 bit** The control is executed via 3 independent 1-bit objects  481 [FC1] Fan speed 1 < 1 = On / 0 = Nothing 482 [FC1] Fan speed 2 < 1 = On / 0 = Nothing 483 [FC1] Fan speed 3 < 1 = On / 0 = Nothing **Yes, custom:** The speed control and the operation modes can be customized with up to 5 1-bit objects  481 [FC1] Fan custom 1 < On / Off 482 [FC1] Fan custom 2 < On / Off 483 [FC1] Fan custom 3 < On / Off 484 [FC1] Fan custom 4 < On / Off 485 [FC1] Fan custom 5 < On / Off This option shows an additional tab to configure each one of the 5 objects in "*Fan manual -> Fan Manual custom*"	
Allow manual mode changeover by object	**Only with Auto/Manual object** Auto/Manual object & Manual Fan objects

Only with Auto/Manual object: Switching to Manual mode only with this object is possible Auto/Manual object & Manual Fan objects: Switch to Manual mode with the Auto/Manual object and also with any other object that allows switching the fan speed. *Note:* If the speed is switched manually, the system will switch to manual mode.	
Temporized Manual Fan control, return to Auto after: - **Base: 1h** (1 min) - **Factor: 1…255** -	**No** Yes
The Manual mode can be enabled with a timer here. If Manual mode is activated, after completion of the defined time, the system goes back to Auto mode.	
Attention! Fan speed operation mode 1 – "Deny Fan OFF manual" is activated in default parameters. To allow Fan OFF, the restriction should be disabled or changed.	

5.5 Valve

Parameter	Settings
The following parameters are available to configure the valve options	
Type of valve	**NC (0%=Close, 100%=Open)** NO (100%=Close, 0%=Open)
Use this parameter option to set whether the output valve closes with 0% and opens with 100% or if it closes with 100% and opens with 0% values.	
Type of control signal The options of this parameter will depend of the "*Type of Fan Coil*" parameter selected in "*Fan Coil Settings"* tab	
Type of control signal (Type of Fan Coil = Heat (2 pipes) or Type of Fan Coil = Cool (2 pipes))	**1 byte PI value** 1 bit PWM
The following options are available when the fan coil type selected in "Fan Coil Settings" is: Type of Fan Coil = Heat (2 pipes) or Type of Fan Coil = Cool (2 pipes) **1 bit PWM:** The valve is controlled via 1 bit DPT values **1 byte PI value:** The valve is controlled via 1byte scaling DPT values For the "1 byte PI value" selection, the following additional options appear:	
PWM cycle time Factor	1 min 15
The total PWM cycle duration is 15 minutes with default values	
Type of control signal (Type of Fan Coil = Heat/Cool (2 pipes))	**1 bit PWM (common Heat/Cool obj)** 2 x 1 bit PWM (common Heat/Cool obj) 1 byte PI value (common Heat/Cool obj) 2 x 1 byte PI value (common Heat/Cool obj)
1 bit PWM (common Heat/Cool obj): The valve is controlled via a single 1 bit object for the Heat/Cool modes **2 x 1 bit PWM (individual Heat/Cool obj):** The valve is controlled via two individual 1 bit objects for the Heat/Cool modes **1 byte PI value (common Heat/Cool obj):** The valve is controlled via a single 1 byte object for the Heat/Cool modes **2 x 1 byte PI value (individual Heat/Cool obj):** The valve is controlled via two 1 byte objects for the Heat/Cool modes All options support the corresponding valve status objects. For the "1 byte PI value" and "2 x 1 byte PI value" selection, the following additional options appear:	
PWM cycle time Factor	1 min 15
The total PWM cycle duration is 15 minutes with default values	

Advanced functions	**No** Yes	
The advanced functions linked to the valve allow for additional control functions.		
Time to close the valve (from 100% to 0%)	**No** Yes	
Time to close the valve		
Base Factor	**1 min** 1	
The default time for valve closure by the system is 1 minute.		
Minimum the valve must remain open.	**No** Yes	
The time in which the valve must remain open when the system opens. The configured time must have elapsed before the status can be changed to "Closed".		
Minimum time the valve must remain open	**No** Yes	
Define here the time in which the valve must stay open when the system opens it. The configured time must elapse before it can change its status to closed.		
Base Factor	**1 min** 1	
The default time for the valve to stay open, when opened by the system, is 1 minute.		
Minimum frequency to allow valve changes	**No** Yes	
Note! After activation, the valve will not accept a new activation. The last value remains active for the configured time.		
Base Factor	**1 min** 2	
The default time during which the valve will not accept any changes is 1 minute.		
When changing heating / cooling, the valve remains closed for	**No** Yes	
Base Factor	**1 min** 1	**1 min** 1
Cyclic sending of valve output	**No** Yes	
The valve's status values can be sent to the bus cyclically.		
Base Factor	**1 min** 1	
The default time for cyclic sending is 1 minute.		
Thermostat monitoring: Valve behaviour	**No** Set value Execute alarm 1 Execute alarm 2 Execute alarm 3 Execute alarm 4 Execute alarm 5 Execute alarm 6 Execute alarm 7 Execute alarm 8	
It shows the parameters to establish the valve operation when the thermostat monitoring function causes an error.		

The following options are available: **No:** No action takes place on the valve and the parameters are hidden. **Set value:** Set the configured value on the valve. **Execute alarm 1..8:** The configured behaviour will be executed on the selected alarm under "Advanced functions -> Alarms -> Alarm X" The following parameter is shown when the option “Set value” has been selected:	
Valve value on error	**0%**
Determine here the positioning value while an error is detected in the thermostat monitoring.	
Behaviour when monitoring error ends	**Set to tracked state**
The valve keeps the position it should have (had there been no error)	
Purge valve (removes air & calcification)	**No** Yes
This function avoids eventual blocking of the valve due to the calcification caused when there is no water flow during long periods of time (valve not in use). Establish here the value for valve opening, duration time and frequency. This function might be enabled from the corresponding “Purge valve” object.	
Duration: Valve remains open during Factor	**1 min** **10**
Define here the time during which the valve will remain in the configured position.	
Frequency (valve opens every)	**Weeks** Only by object Minutes Hours Days Months
Determine here how often the valve positioning will be enabled and how long (time set in the “Duration” parameters) The options available are as follows: **Weeks** The base value will be set in weeks **Only by object**: The activation will only be done via the communication object intended for this purpose. **Minutes**: The base value will be set in minutes **Hours:** The base value will be set in hours **Days:** The base value will be set in days **Months:** The base value will be set in months	
Factor	**1**
Valve position	**100%**
The positioning value of the valve can be configured here when the function is enabled.	

5.5.1 Alarms valve (Valve tab)

Parameter	Settings
Forced valve position on alarm 1..8	**Nothing** Actual position Set to position
Decide here the behaviour of the valve when enabling each one of the 8 alarms already existing in the "*General settings -> Advanced functions-> Alarms"*. The following options are available as long as the selected alarm is active: **Nothing:** No action takes place **Actual position:** The valve position is forced to be the current active position **Set to position:** The valve position is forced to be the value established in the parameter "valve position" Attention!! Priorities: Alarm 1 (highest)...8 (lowest)	
Unforced fan speed at end of all alarms	**Set to tracked state**
Set to tracked state: The position of the valve is set to match the one that it should have had if no alarm have been triggered.	

5.6 Status

Parameter	Settings
Show or hide the status objects of the different functions available to the fan coil device.	
Trigger object to send all status telegrams	**Yes, with ON** Yes, with OFF Yes, with Both
This option enables the object “Send all status”, which allows forcing the sending of all status values in the fan coil module when the established values are received as follows: **Yes, with ON:** The sending will be forced when the value ON is received **Yes, with OFF:** The sending will be forced when the value OFF is received **Yes, with Both:** The sending will be forced when both the values ON and OFF are received	
Heat/Cool status	No **Yes**
The status object to indicate the current Heat/Cool mode is enabled	
Fan Coil On/Off status	No **Yes**
This option is shown when the “Fan Coil settings -> On/Off object” has been previously activated The status object to indicate whether the fan coil module is ON or OFF is enabled	
Fan speed status	**1 byte enumerated status** 1 byte scaling status Both Custom
The status object type to learn the fan speed can be selected here: **1 byte enumerated status:** The 1 byte DPT 5.010 counter pulses object is enabled The sending values are as follows: Speed 0 = 0, Speed 1 = 1, Speed 2 = 2, Speed 3 = 3 **1 byte scaling status:** The 1 byte DPT 5.001 percentage object is enabled. The sending values are as follows: Speed 0 = 0%, Speed 1 = 33%, Speed 2 = 67%, Speed 3 = 100% **Both:** Both objects above mentioned are simultaneously enabled **Custom:** The representation of the current fan value can be fully customized: When the option “Custom” is selected, the following options are available: 1 byte Fan Speed status object 1 bit Fan Speed status object 1 byte free allocable status values	
1 byte Fan Speed status object	No **Yes**
The two 1 byte objects are shown or hidden: - 1 byte enumerated status - 1 byte scaling status	
1 bit Fan Speed status object	**No** Yes
Four 1 bit objects can be individually enabled, one for each fan speed. The following options are shown:	
Fan Off. 1 bit status object	**No** 1 = Fan Off, 0 = X 1 = Any speed active, 0 = Fan Off

No: The Fan OFF status object is hidden **1 = Fan Off, 0 = X:** It indicates speed 0 with the ON value. With value OFF, it indicates that a speed different to 0 is enabled **1 = Any speed active, 0 = Fan Off:** With value ON, it indicates that a speed different to 0 is enabled With value 0, it indicates that speed 0 is enabled	
Speed 1. 1 bit status object	No Yes
The 1 bit object that indicates the fan speed 1 is shown or hidden - 1 value = ON - 0 value = Nothing	
Speed 2. 1 bit status object	No Yes
The 1 bit object that indicates the fan speed 2 is shown or hidden - 1 value = ON - 0 value = Nothing	
Speed 3. 1 bit status object	No Yes
The 1 bit object that indicates the fan speed 3 is shown or hidden - 1 value = ON - 0 value = Nothing	
1 byte free allocable status values	No Fan speed 1 byte unsigned Fan speed Man + Fan speed Auto
The status values can be freely customized for each one of the fan speeds. The available options are: **No:** Parameters are hidden **Fan speed 1 byte unsigned:** Customize values for 1 byte unsigned **Fan speed Man + Fan speed Auto:** Independent values can be customized according to the enabled fan mode (4 values for Manual and 4 values for Auto, independent from each other).	
The following values have been set by default for the **Fan speed 1 byte unsigned** object - Fan Off status - Speed 1 status - Speed 2 status - Speed 3 status	**Values** 0 1 2 3
Customized values can be defined with the purpose of meeting the requirements of the different visualization solutions available in the market.	
The following values have been set by default: **Fan speed Man + Fan speed Auto** object - Fan Off (Manual mode) status - Speed 1 (Manual mode) status - Speed 2 (Manual mode) status - Speed 3 (Manual mode) status - Fan Off (Auto mode) status - Speed 1 (Auto mode) status - Speed 2 (Auto mode) status - Speed 3 (Auto mode) status	**Values** 0 1 2 3 4 5 6 7
Note: Intended for a single (multi-status) element to show both the actual speed & mode selection. i.e (Value 1 = Speed 1 in Manual mode); (Value 5 = Speed 1 in Auto mode)	

Special mode status	No **Yes**	No **Ja**
Output valve status heating	No **Ja**	
Output valve status cooling	No **Ja**	
Request heating status	No **Ja**	
Request cooling status	No **Ja**	
Automatic / Hand status	No **Ja**	
Status valve purge	No **Ja**	

Here the status objects of the listed functions can be activated / deactivated
Yes = Active
No = inactive		

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 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.

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 staring for 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, 4?	

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 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.	

6.1.2 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.	

6.1.3 Analog

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 “rigged” 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 Logics

There are 25 logic functions available in Power Block o16 and 35 in Power Block o8

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 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 *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 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 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 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 *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 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.	

6.2.3.1 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 *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 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 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 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 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.	

6.2.5.1 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 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 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	

6.3.1 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.	

6.4 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.	

6.4.1 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.	

6.4.1.1 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.	
Factor changeable by object / Remaining time cyclic sending	**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. Additionally, to the above function, when the timer is active, this object will send the total remaining time up to 10 times with steps of 10% of the total time value until the timer finish. In order to disable this function, the “T” flag must be deactivated.	
Advanced staircase function	**No** Yes
Here the advanced functions can be activated.	

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 ON telegrams received. This resulting time will never exceed the parameterized maximum staircase time in the option “Maximum staircase time Base/Factor” It is important to keep in mind that the multiplication will only be done starting from the first triggering telegram	

(so, the Multiplying staircase function will only be executed when starting the staircase, not during execution). Therefore, these ON telegrams may not be longer than 1 second apart. 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).

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). But this function will only be executed after more than 1 second has elapsed between the triggering events of the timer (if less than 1 second, see behaviour in section MULTIPLY STAIRCASE).

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.

The channel, 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.

6.4.1.2 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.	

6.5 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)	

6.5.1 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.	

6.5.2 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”

6.5.2.1 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 to not exceed the appointed maximum ¼ hour energy values and therefor reduce the monthly costs.	

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 by 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	
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".	

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.	

6.5.3 Setpoints 4 … 10

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"	

6.5.3.1 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 to not exceed the appointed maximum ¼ hour energy values and therefor reduce the monthly costs.	

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".	

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.	

6.6 *Facade Control*

Parameter	Settings
Facade Control	**No** Yes
Here the Facade Control can be activated. Facade control function can be used to control the different shutter/blind channels from a weather station for automatic shading control, all of them ordered by group of facades. Up to a maximum of 4 groups will be possible to associate the channels, classified by the next default text descriptions: North, South, East, West. When Facade control is active, all the individual channel slats/blind position objects will be inactive (**the objects connected to the individually push buttons**), so the channels will only react using the Facade control objects. Additionally, this function can be deactivated temporary/manually, where in such a case, all the channel slats/blind position objects will be meanwhile activated in order to enable again the individually shutter/blind push buttons functionality. Channel alarm function has highest priority to Facade control objects.	

6.6.1 Facade 1..4

Parameter	Settings
Facade 1 description	**Text**
Facade 1 … Facade 4	**No** Yes Yes, temporized
When selecting **“No”**, all the parameters are hidden When selecting **“Yes”**, the Facade Control objects are shown. When selecting **“Yes, temporized”** is possible to set the time to change back to automatic mode when the object is active with value 1.	
Time to change back to automatic mode	**1h**
Behaviour when exiting Facade control	**Do nothing** Move Down Move Up Move to blind position Move to slat position Move to slat and blind position Move to preset Set to tracked state
The “Behaviour when exiting Facade control” will be executed when the object “Facade X Auto/Manual” receives the value 0.	
Reaction on bus voltage failure	**Don’t execute anything** Same as blind channel behaviour

It is possible to set an action to the complete group of shutter/blind channels when the bus voltage fails. **Don't execute anything:** The channels will not do any action when bus voltage fails. **Same as blind channel behaviour:** Each channel will execute the behaviour configured individually in the "Reaction on bus voltage failure" parameters when bus voltage fails.	
Reaction on bus voltage recovery	**Don't execute anything** Same as blind channel behaviour
It is possible to set an action to the complete group of shutter/blind channels when the bus voltage is recovered. **Don't execute anything:** The channels will not do any action when the bus voltage is recovered. **Same as blind channel behaviour:** Each channel will execute the behaviour configured individually in the "Reaction on bus voltage failure" parameters when the bus voltage is recovered.	

Parameter	Settings
Allocation of Channel A, B, and C	**No** Facade 1 Facade 2 Facade 3 Facade 4
Here it is possible to include each shutter/blind channel individually into each Facade group. A maximum of 4 Facades are available to include the shutter/blind channel. **Attention!** The specific shutter/blind channel only appears into the allocation section of this tab, when it is configured as a shutter/blind channel into "General Settings -> Outputs" tab.	
Object to exclude Ch.A..C from facade	**No** Yes Yes, temporized
No: The object Facade Exclude Ch.A…C is hidden. **Yes:** It is possible to exclude a specific shutter/blind channel from the Facade Control function sending a value 0 to the object "Facade Exclude Ch.A…C" (Manual mode) To include it again into the Facade Control group, a value 1 must be set in the object (Automatic mode) **Yes, temporized:** It is possible to exclude a specific shutter/blind channel from the Facade Control function sending a value 1 to the object "Facade Exclude Ch.A…C temporized". To cancel the temporization, a value 1 must be set in the object.	
Time to change channel to automatic mode	**1h**
The manual mode will be activated during the time established in this parameter. After this time, the channel will be changed to Automatic mode into the Facade control group.	

Parameter	Settings
Weather station monitoring	**No** Yes
If this function is activated, the Facade control objects will be monitored in order to detect if these objects are receiving periodically values into the period time configured in the next parameter. An alarm will occur if no slat/blind position telegram is received (i.e. because a faulty weather station). The alarm will be activated by sending a telegram with value 1 via the object “Facade monitoring alarm”. The alarm will be finished when the Facade control objects start to receive again the values into the period time. By using the same object, when the alarm is inactive, a telegram with the value 0 will be sent.	
Monitoring time base	**5 min**
This is the period where the objects slat/blind position will be monitored. They must receive their telegram into this time to keep inactive the alarm.	
Behaviour when alarm occurs	**Do nothing** Do exiting behaviour
Do nothing: In case of the alarm is activated the Facade control will do not anything. **Do exiting behaviour:** In case of the alarm is activated, the exiting behaviour will be executed and the individual slats/blind positioning objects will be activated again in order to have the control from the individual push buttons.	

6.7 *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, and 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 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	

6.7.1.1 Input object

Parameter	Settings
Output object to send variable	General **Switching channels** Blind channels Logic 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 sub-filter 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
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
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 35
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	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.	

6.7.1.2 Output 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
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
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.	

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	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.	

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 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	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 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.	

6.9 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.	

6.9.1 ADVANCED FUNCTIONS

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.	

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 / 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.	

Parameter page: ADVANCED FUNCTIONS / Advanced scenes / Overwrite individually

Parameter	Settings
Advanced scenes	Overwrite individually
- First scene … - Tenth scene	Overwrite Don’t overwrite
Select here whether to overwrite or not	

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.	

Parameter page: ADVANCED FUNCTIONS / Timers / Overwrite individually

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

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.	

Parameter page: ADVANCED FUNCTIONS / Setpoints / Overwrite individually

Parameter	Settings
Setpoints	Overwrite individually
- Setpoint 1 … - Setpoint 10	Overwrite Don’t overwrite
Select here whether to overwrite or not	

6.9.1.1 Enduser Parameter 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.	

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	

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	

6.10 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.	

6.11 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.	

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 Power Block io64 device the file would be: P3_io64.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 programming 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**.

9 ANNEX

9.1 *Annex 1: Manual Control (Parameter Mode)*

The **outputs** of the actuator have 2 push buttons and 2 status LEDs for each output channel on the front side. These buttons can be activated to control each and every channel/output individually if you select “yes” in the relevant parameter options in Binary outputs and/or Shutter/Blinds.
The LEDs represent:
For Binary outputs: The top row: channels A1, A2, B1, B2.
For Shutter/blinds: The top row: channel’s first relay A1->UP, A2->DOWN, B1-UP, etc.
The **inputs** of the actuator have 1 push button and 1 status LED for each input on the below LED row
These buttons can be activated to control each and every input individually if you select “yes” in the relevant parameter options in Binary Input.
The LEDs represent: The below row inputs 1&4, 2&5, 3&6 actual input status

9.1.1 PARAMETER MODE

<table>
<tr><td colspan="2"><b>Manuel Control – Parameter Mode</b>

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>BINARY</td><td>SHUTTER/BLIND</td></tr>
<tr><td>Press action: Sends Toggle ON/OFF command “0/1” to the “Switching” object

LED = ON (indicates channel status)
LED = OFF (indicates channel status)</td><td><u>Long press action (Channel output 1)</u>: Sends a UP command “0” to the “Move” object.
<u>Long press action (Channel output 2)</u>: Sends a DOWN command “1” to the “Move” object.

<u>Short press action (any output)</u> (while shutter/blind is moving) of same button: sends a Stop command to the “Stop…” object.

LED blinks while moving UP/DOWN during parameterized time</td></tr>
<tr><td colspan="2"><b>BINARY INPUT</b></td></tr>
<tr><td colspan="2"><u>Press action on 1&4, 2&5, 3&6</u>: Sends Toggle ON/OFF command 0/1 to the “associated object” of the input (simulates the close/open action on the binary contact)
LED = ON (indicates input status -> Input contact closed)
LED = OFF (indicates channel status -> Input contact open)
“Man” push button in the right side for selection inputs status range between input 1..3 (LED = OFF) and inputs 4..6 (LED = Blinking)</td></tr>
</table>

9.1.2 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. 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, any button can be used depending of the channel configuration:

- If “Binary” channel is configured: Press any button for at least 500ms
- If “Blind” channel is configured: Press the two buttons of any channel at the same time for at least 500ms

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).

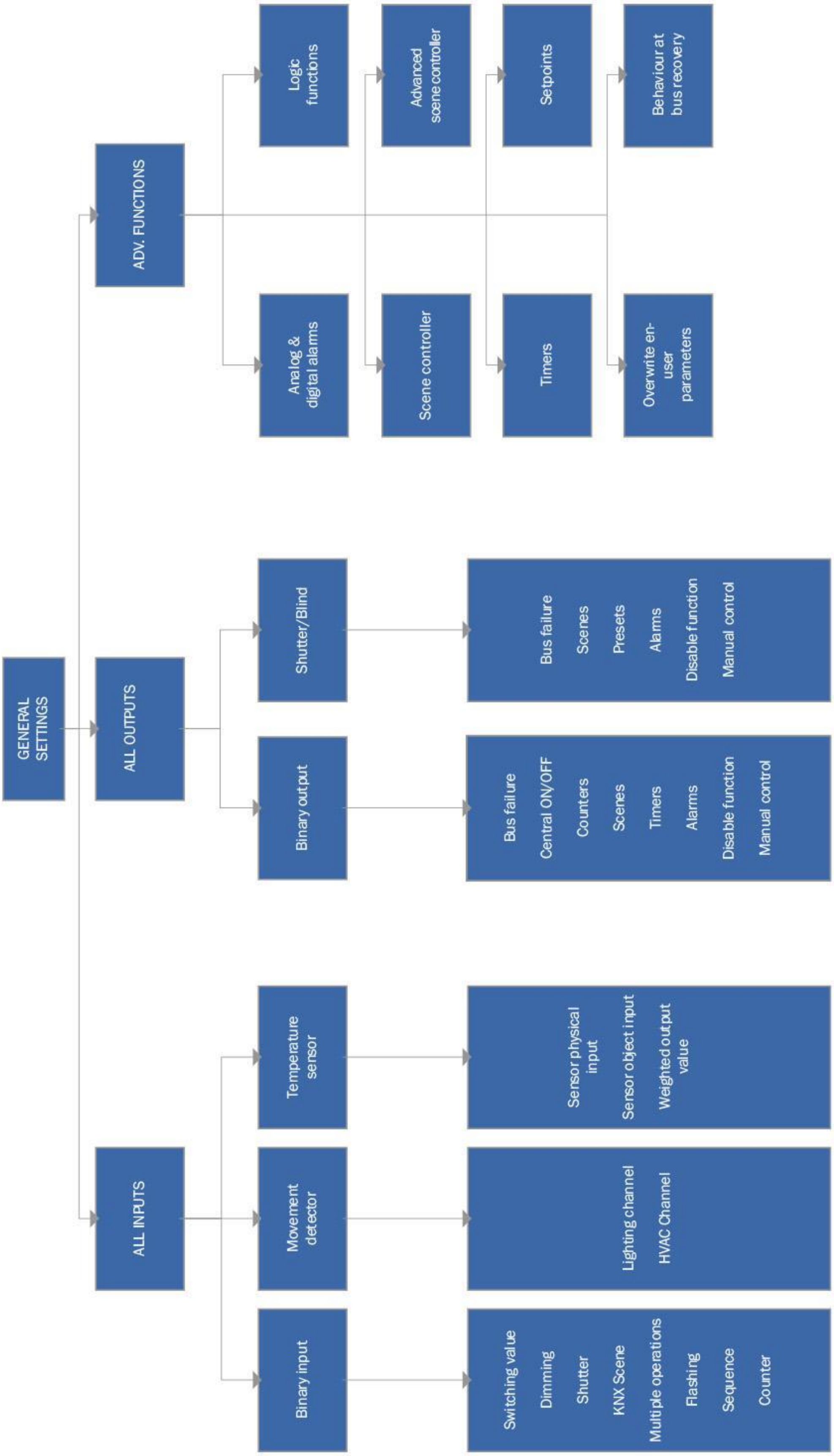
The Action switches/moves the channel, as you can see in the table below:

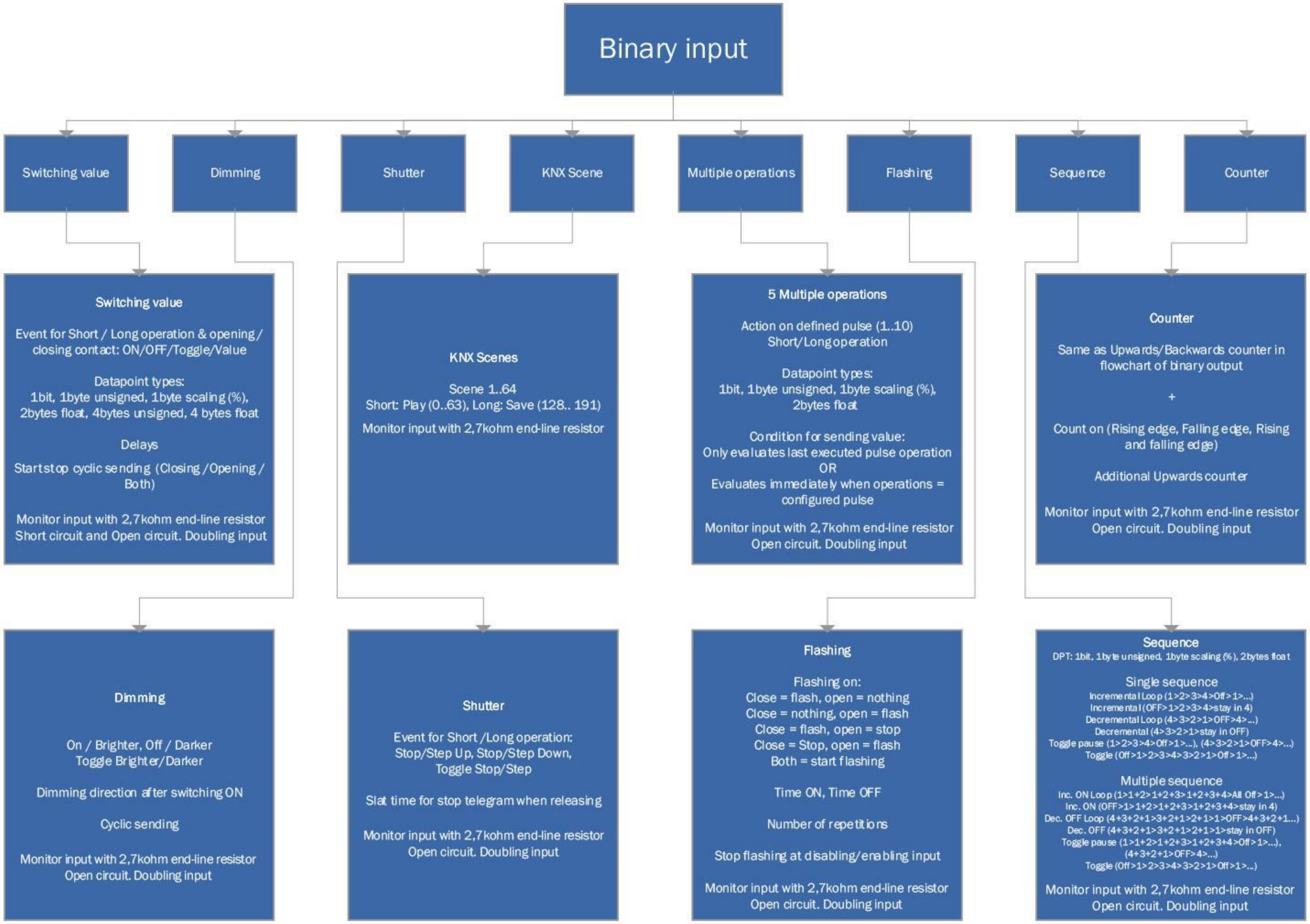
BINARY	SHUTTER/BLIND

Press action: Sends toggle ON/OFF command to the relay (ON = Contact closed / OFF = Contact open)
LED = ON (indicates channel status)
LED = OFF (indicates channel status)
	Rising edge press action (Channel X): Contact closed
	Falling edge press action (Channel X): Contact open
	LED = ON (indicates channel status)
	LED = OFF (indicates channel status)

BINARY INPUT
Don’t apply

9.2 Annex 2 Flowchart





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

Startstop cyclic sending (Closing / Opening / Both)

Monitor input with 2,7kohm end-line resistor
Short circuit and Open circuit. Doubling input

KNX Scenes

Scene 1..64
Short: Play (0..63), Long: Save (128.. 191)

Monitor input with 2,7kohm end-line resistor

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

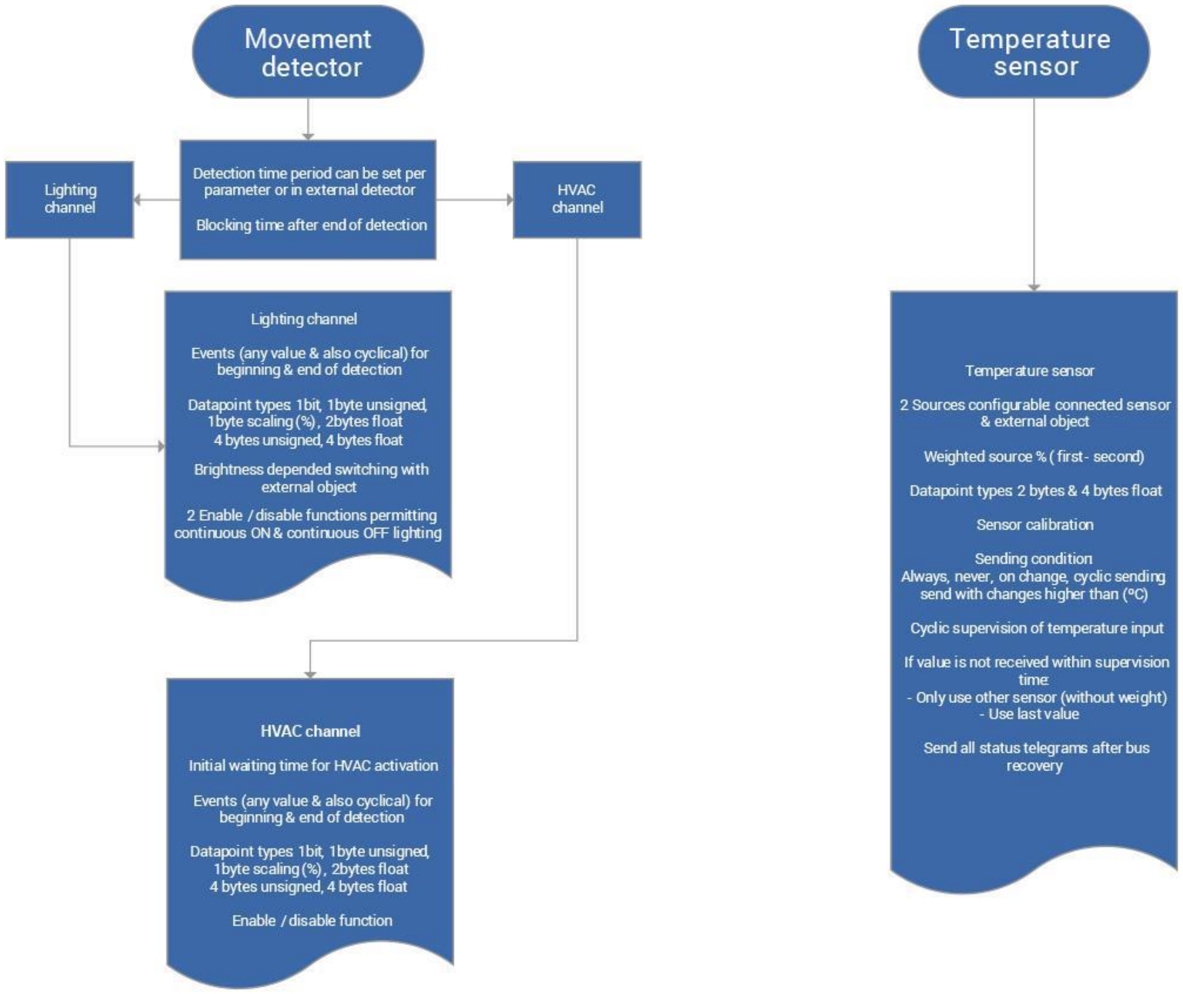
Monitor input with 2,7kohm end-line resistor
Open circuit. Doubling input

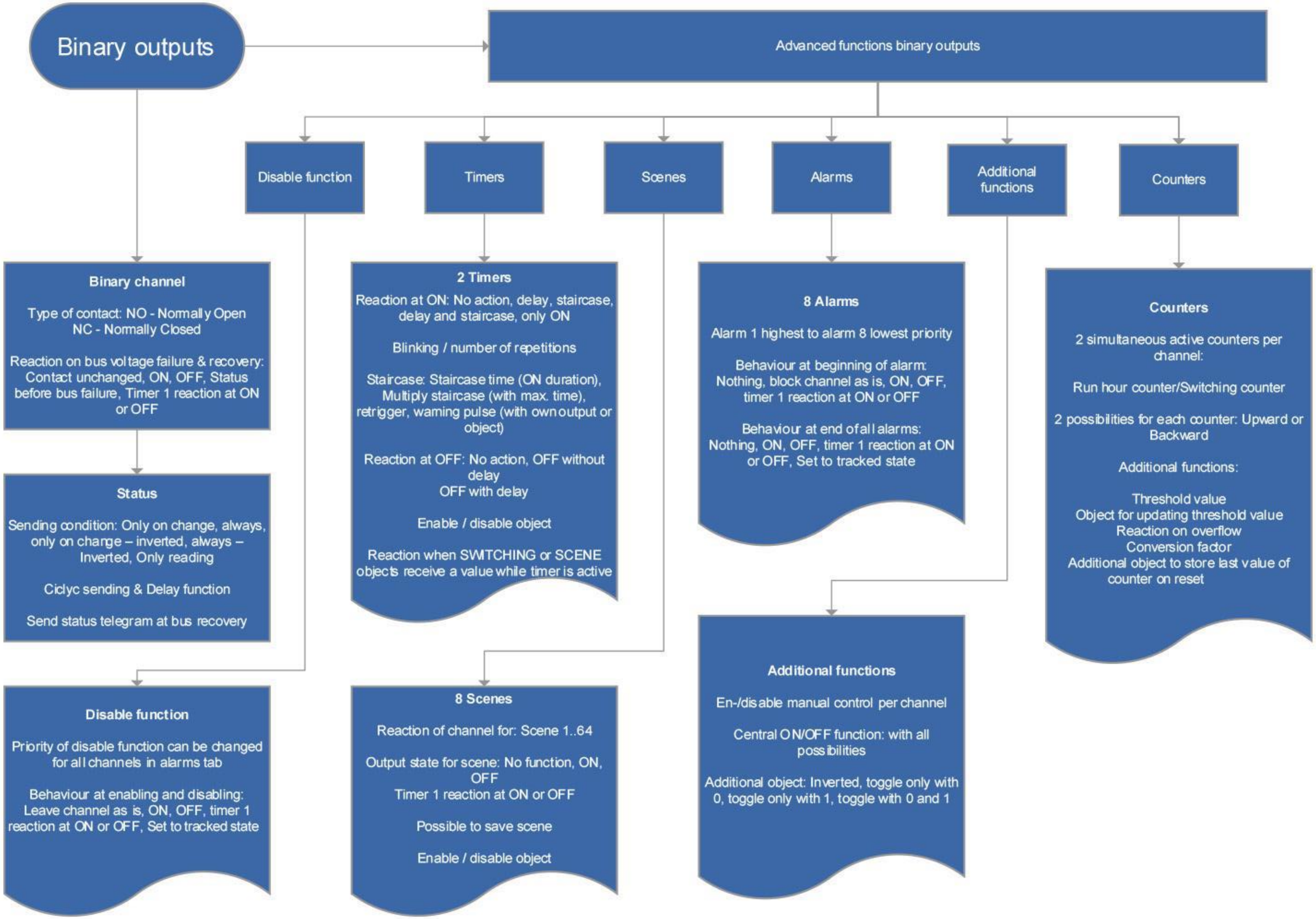
Sequence
DPT: 1bit, 1byte unsigned, 1byte scaling (%), 2bytes float

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 (ºC)
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 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

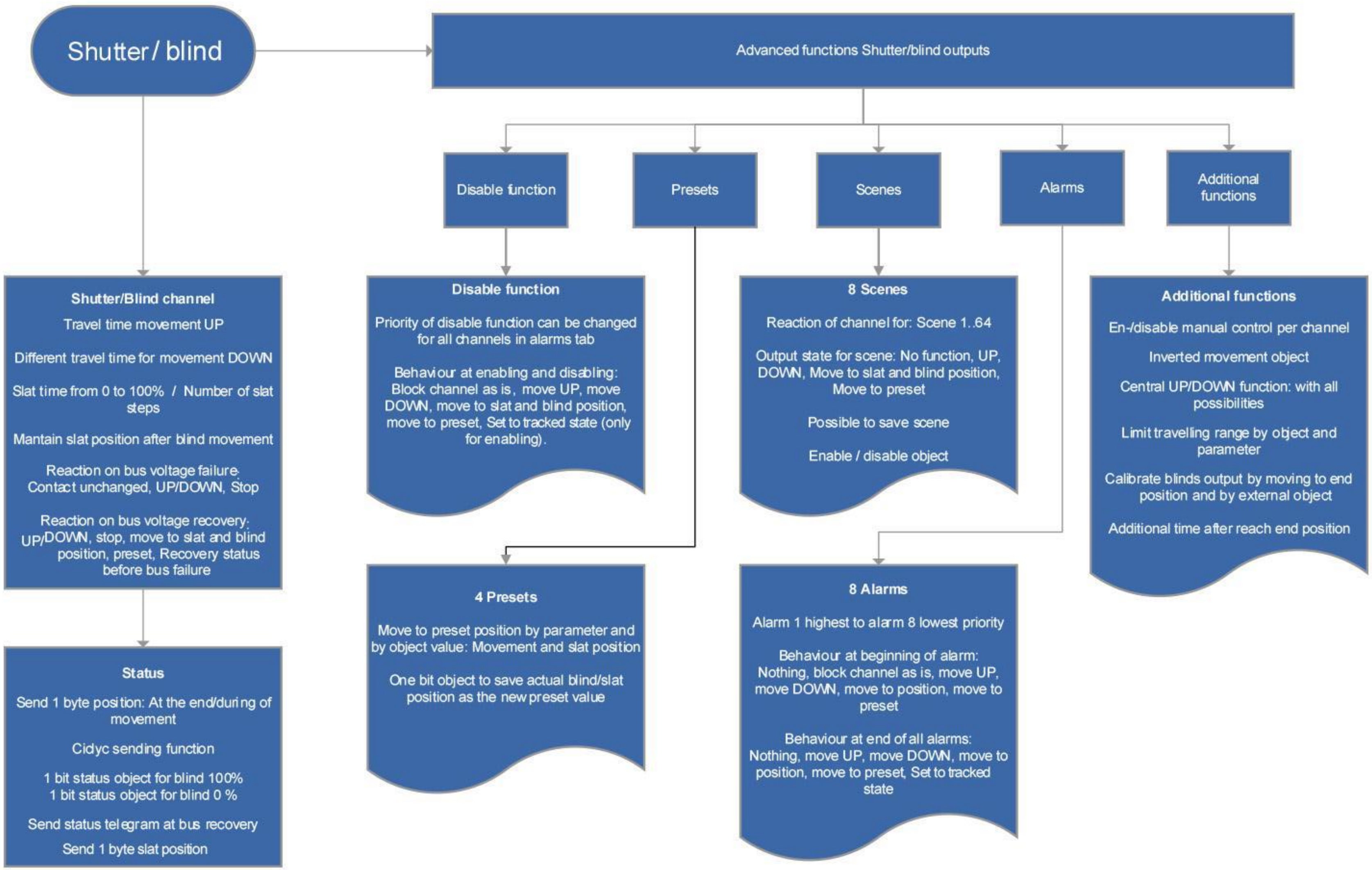
2 simultaneous active counters per channel:

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

Cidyc 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

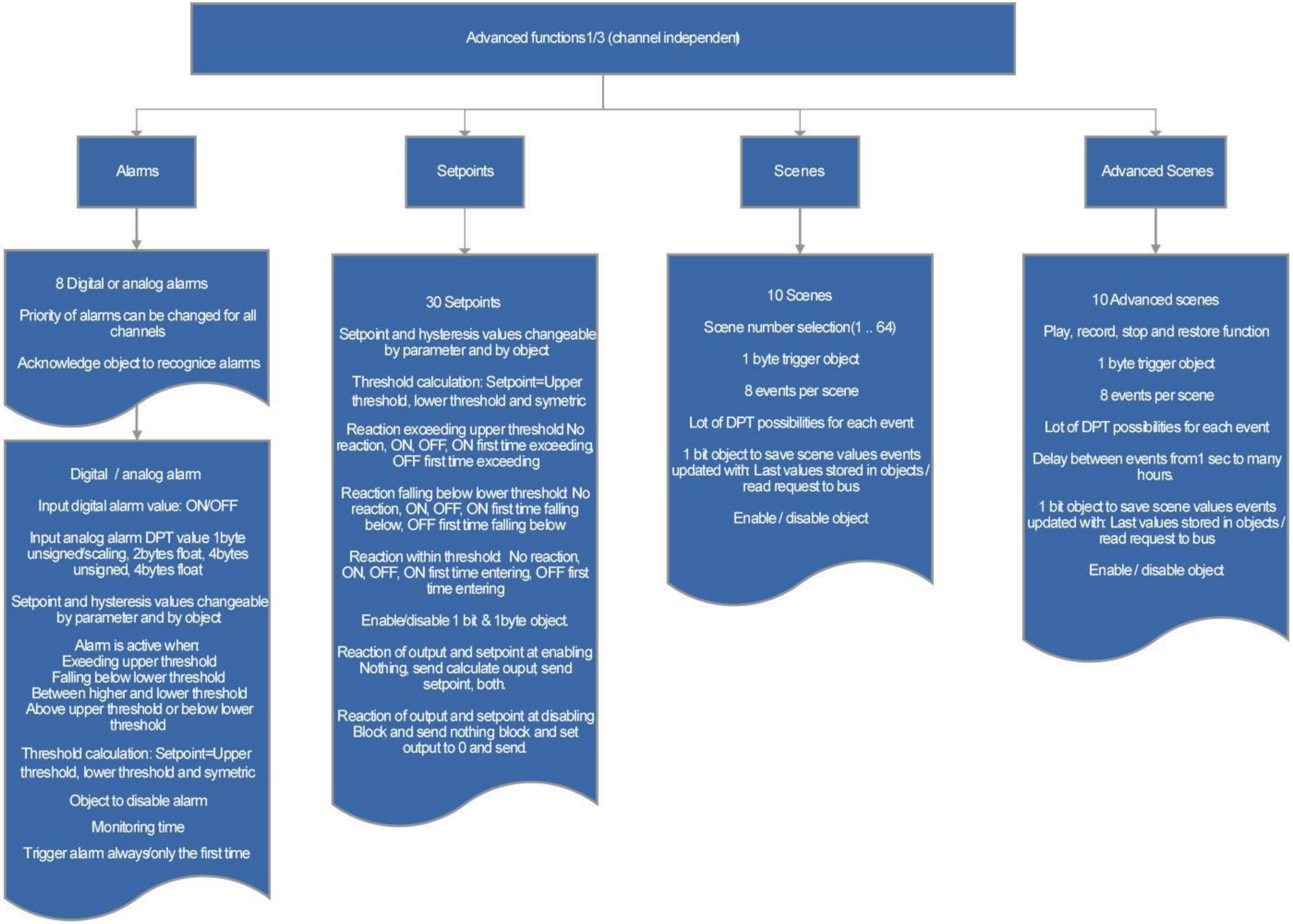
Inverted movement object

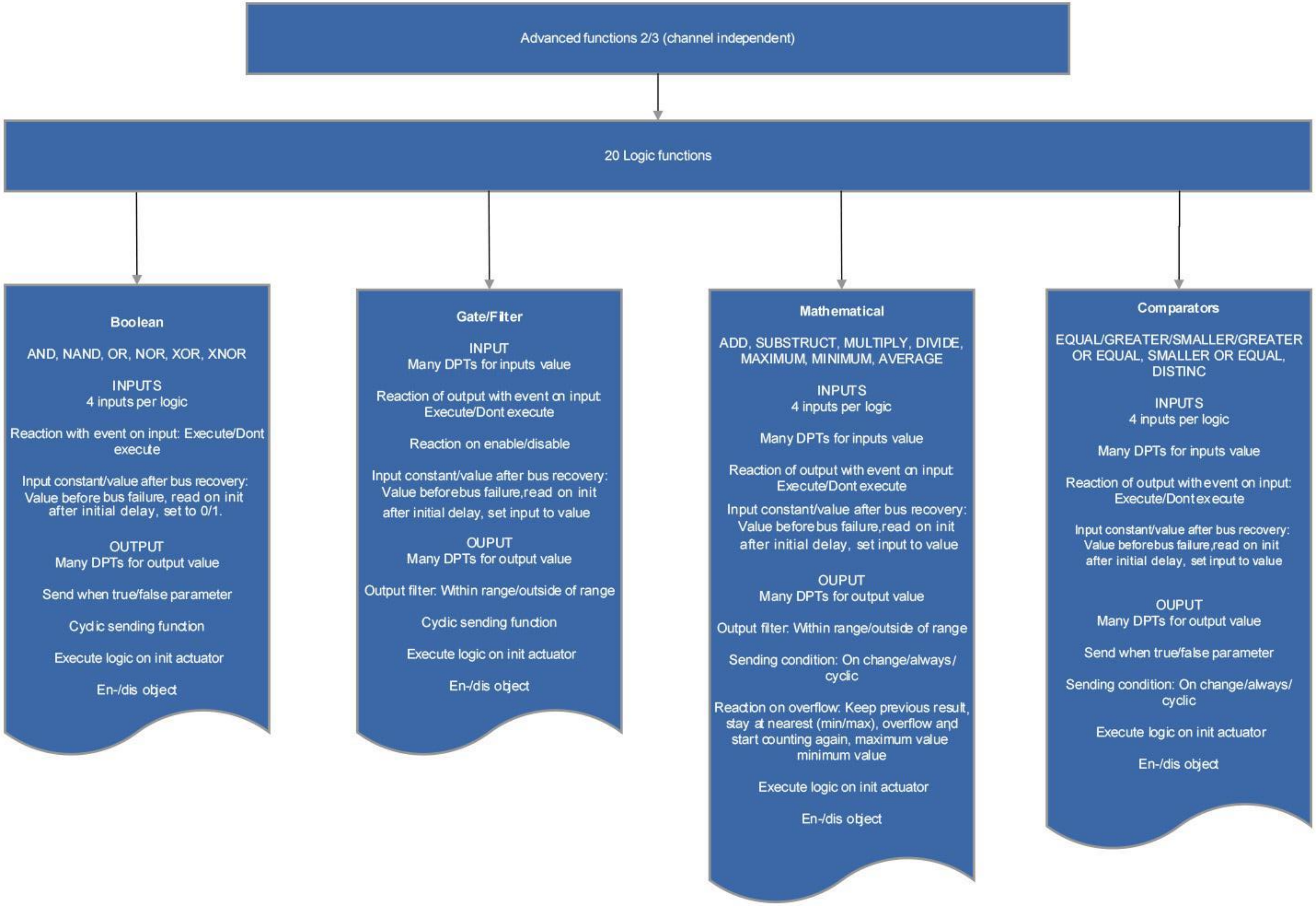
Central UP/DOWN function: with all possibilities

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 functions1/3 (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 from1 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

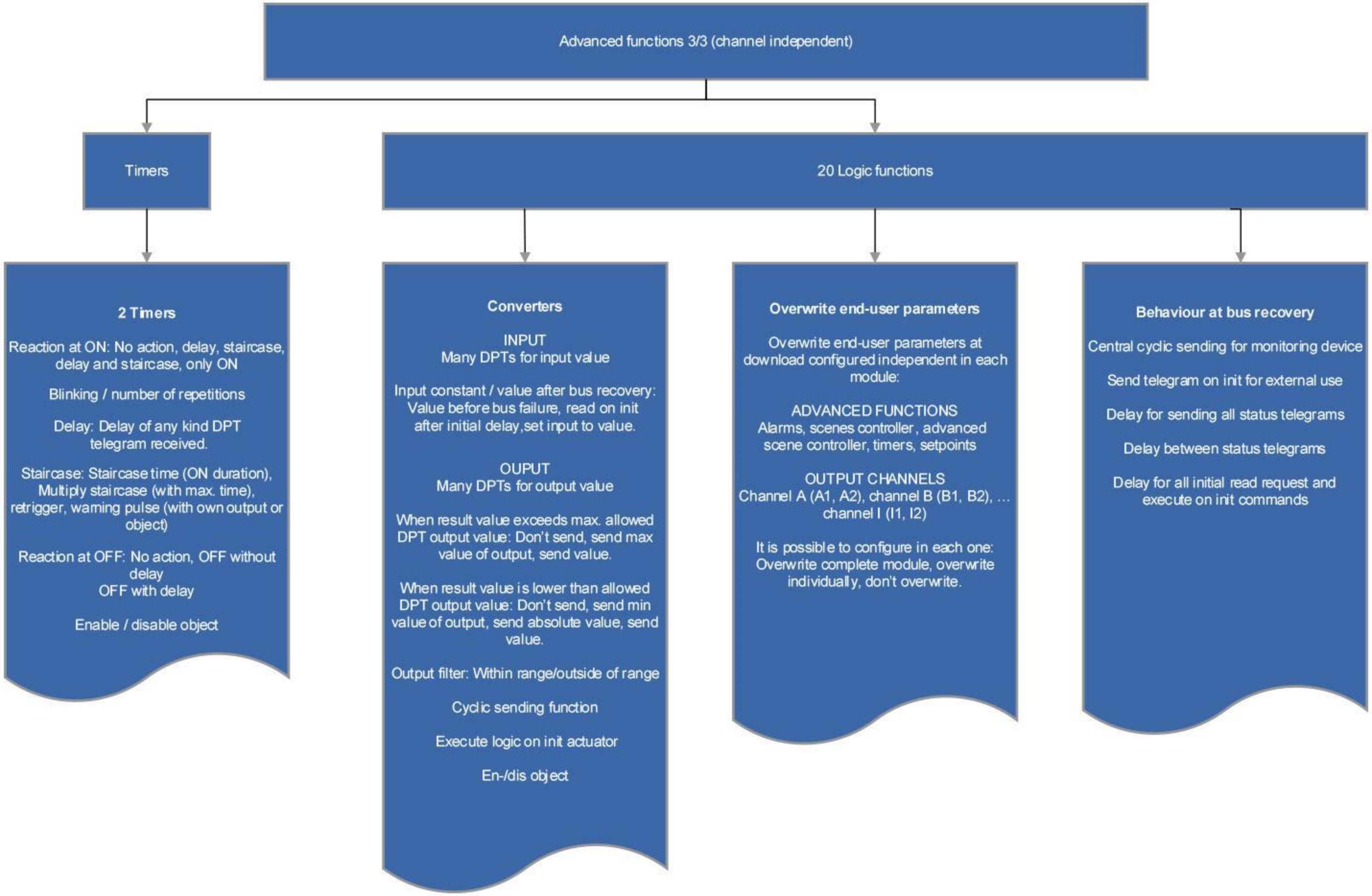
OUPUT
Many DPTs for output value

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