Shutter Module 4 OUT 12-24 V DC

SA04K01KNX

User Manual

Product:	**SA04K01KNX**
Description:	KNX Shutter Module
Document Version:	**1.0**
Date:	27/05/2025

Eelectron SpA, Via Monteverdi 6,
I-20025 Legnano (MI), Italia
Tel: +39 0331.500802 Fax:+39 0331.564826
E-mail: info@eelectron.com Web:www.eelectron.com

C.F. e P.IVA 11666760159
Capitale sociale: 977.400,00€ interamente versato
Tribunale di Milano 359157-8760-07
CCIAA Milano 1486549
Registro AEE: IT18090000010747

INDEX

Disclaimer:

Any information contained in this manual may be changed without notice.
This manual can be freely downloaded from the website: www.eelectron.com
Despite the correctness of the data contained within this document has been verified, it is not possible to exclude the presence of errors or typos; Eelectron therefore assumes no responsibility in this regard. Any corrections that will be necessary will be included in the updates of this manual.

1 – General product information

Installing the application program

The application for the SA04K01KNX 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. After the import, the product can be integrated as usual into the ETS. It can be found under product family “Output Module“ and product type “Shutter/Motor 4-Output DC”.

1.1 Using the application program

Product family:	Output Module
Product type:	Shutter/Motor 4-Output DC
Manufacturer:	Eelectron
Name:	Shutter DIN Module 4 OUT 12-24 V DC
Order number:	SA04K01KNX
N° of communication objects:	998

1.2 Preliminary basic concepts

Output: channel type selection
The outputs of the SA04K01KNXare divided into DC shutter channels. Each channel consists of 4 connectors: 2 of them are mecanichal relays for the shutter/blind control movement and the other 2 are 2 inputs for its respective DC power supply.

The channel type is preselected to be a“Shutter/Blind DC” channel, so the relay distribution for the first channel is as follows:

- Connector 1 (+): Power supply DC+`input
- Connector 2 (-): Power supply DC- input
- Connector 3 (Up): Shutter/blind UP output
- Connector 4 (Down): Shutter/blind DOWN output

When the move action is executed by the channel, the corresponding DC polarity is applied to channel output, depending of the movement direction:

- UP movement: Output 3 = DC+ , Output 4 = DC -
- DOWN movement: Output 3 = DC - , Output 4 = DC+

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 find the Scene Controller.
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

These actuators communicate via the KNX bus based on powerful communication stacks. A total of 998 communication objects for the SA04K01KNX are available for communication.

	Text	Function text	Object Size	Flags	Datapoint type
1	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 b e 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.					
197	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.					
197	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.					
197	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.					
197	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.					
197	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.					
197	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.					
197	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.					
197	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.					
197	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.					

197	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.					
198	Scene 1 disable	< Disable = 1 / Enable = 0	1 Bit	RWC---	[1.003] DPT_Enable
The scene can be disable with a 1					
198	Scene 1 disable	< Disable = 0 / Enable = 1	1 Bit	RWC---	[1.003] DPT_Enable
The scene can be disable with a 0					
199	Scene 1 event 1	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the first event for the first advanced scene.					
199	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.					
199	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.					
199	Scene 1 event 1	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the first event for the first advanced scene.					
199	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.					
199	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.					
199	Scene 1 event 1	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the first event for the first advanced scene.					
199	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.					
199	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.					
199	Scene 1 event 1	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the first event for the first advanced scene.					
200	Scene 1 event 2	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the second event for the first advanced scene.					
200	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.					
200	Scene 1 event 2	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the second event for the first advanced scene.					
200	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.					

200	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.					
200	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.					
200	Scene 1 event 2	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the second event for the first advanced scene.					
200	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.					
200	Scene 1 event 2	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the second event for the first advanced scene.					
200	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.					
200	Scene 1 event 3	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the third event for the first advanced scene.					
200	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.					
201	Scene 1 event 3	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the third event for the first advanced scene.					
201	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.					
201	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.					
201	Scene 1 event 3	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the third event for the first advanced scene.					
201	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.					
201	Scene 1 event 3	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the third event for the first advanced scene.					
201	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.					
201	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.					
202	Scene 1 event 4	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the fourth event for the first advanced scene.					

202	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.					
202	Scene 1 event 4	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fourth event for the first advanced scene.					
202	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.					
202	Scene 1 event 4	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the fourth event for the first advanced scene.					
202	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.					
202	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.					
202	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.					
202	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.					
202	Scene 1 event 4	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fourth event for the first advanced scene.					
203	Scene 1 event 5	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the fifth event for the first advanced scene.					
203	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.					
203	Scene 1 event 5	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the fifth event for the first advanced scene.					
203	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.					
203	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.					
203	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.					
203	Scene 1 event 5	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the fifth event for the first advanced scene.					
203	Scene 1 event 5	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the fifth event for the first advanced scene.					
203	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.					

203	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.					
204	Scene 1 event 6	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the sixth event for the first advanced scene.					
204	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.					
204	Scene 1 event 6	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the sixth event for the first advanced scene.					
204	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.					
204	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.					
204	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.					
204	Scene 1 event 6	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the sixth event for the first advanced scene.					
204	Scene 1 event 6	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the sixth event for the first advanced scene.					
204	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.					
204	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.					
205	Scene 1 event 7	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the seventh event for the first advanced scene.					
205	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.					
205	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.					
205	Scene 1 event 7	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the seventh event for the first advanced scene.					
205	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.					
205	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.					
205	Scene 1 event 7	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the seventh event for the first advanced scene.					

205	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.					
205	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.					
205	Scene 1 event 7	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the seventh event for the first advanced scene.					
206	Scene 1 event 8	<> On / Off	1 Bit	-WCTU-	[1.001] DPT_Switch
This is the eighth event for the first advanced scene.					
206	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.					
206	Scene 1 event 8	<> 0..100%	1 Byte	-WCTU-	[5.1] DPT_Scaling
This is the eighth event for the first advanced scene.					
206	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.					
206	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.					
206	Scene 1 event 8	<> 2 bytes float	2 Bytes	-WCTU-	[9] 9.xxx
This is the eighth event for the first advanced scene.					
206	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.					
206	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.					
206	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.					
206	Scene 1 event 8	<> 4 bytes float	4 Bytes	-WCTU-	[14] 14.xxx
This is the eighth event for the first advanced scene.					
297	Timer 1 trigger	< On / Off	1 Bit	-WC---	[1.001] DPT_Switch
This is to trigger the first timer					
297	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)					
297	Timer 1 trigger	< 1 byte scaling	1 Byte	-WC---	[5.1] DPT_Scaling
This is to trigger the first timer (only for delay)					
297	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)					
297	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)					

297	Timer 1 trigger	< 2 bytes float	2 Bytes	-WC---	[9] 9.xxx
This is to trigger the first timer (only for delay)					
297	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)					
297	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)					
297	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)					
297	Timer 1 trigger	< 4 bytes float	4 Bytes	-WC---	[14] 14.xxx
This is to trigger the first timer (only for delay)					
298	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.					
299	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.					
300	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					
301	Timer 1 output	> On / Off	1 Bit	--CT--	[1.1] DPT_Switch
This is the output object of the timer.					
301	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)					
301	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)					
301	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)					
301	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)					
301	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)					
301	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)					
301	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)					

301	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)					
301	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)					
347	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					
348	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					
348	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					
348	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					
348	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					
348	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					
348	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					
349	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)					
350	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					
350	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					
350	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					
350	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					

350	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					
350	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					
351	Setpoint 1 disable	< On / Off	1 Bit	RWC---	[1.003] DPT_Enable
The setpoint can be disabled with this object					
351	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.					
397	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.					
398	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.					
399	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					
399	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					
400	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					
400	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					
413	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.					
414	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.					
414	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.					
418	[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					

419	[A] Stop (Blind=Stop/step)	< 0=stop/step, 1=stop/step	1 Bit	-WC---	[1.007] DPT_Step
This is to stop/step the blind 0=stop/step up, 1=stop/step down					
420	[A] Move to position	< 0..100%	1 Byte	-WC---	[5.1] DPT_Scaling
The blind can be moved to a specific absolute position with this object.					
421	[A] Move slat	< 0..100%	1 Byte	-WC---	[5.1] DPT_Scaling
This object is to move the slats to an absolute position.					
421	[A] Move slit	< 0..100%	1 Byte	-WC---	[5.1] DPT_Scaling
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.					
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.					
422	[A] Change upper limit	<> 0..100%	1 Byte	RWCT--	[5.1] DPT_Scaling
The blinds can have limits configured in the parameters and the upper limit can be changed by using this object. Should an invalid value (upper limit must be smaller than lower limit) be sent to this object it will be rejected and the previous value will be restored and send to the bus.					
423	[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.					
424	[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.					
425	[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.					
426	[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.					
427	[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.					
427	[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.					
428	[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					
429	[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					
430	[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					
431	[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					

432	[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					
433	[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					
434	[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					
435	[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					
436	[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					
437	[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					
438	[A] Preset 3 change slat position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute slat position which will be set when calling preset 3					
439	[A] Preset 4 change slat position	< 0..100%	1 Byte	RWC---	[5.1] DPT_Scaling
This is to change the blind absolute slat position which will be set when calling preset 4					
440	[A] Preset 1 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
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					
441	[A] Preset 2 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
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					
442	[A] Preset 3 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
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					
443	[A] Preset 4 save	< 1 = Save, 0 = Nothing	1 Bit	-WC---	[1.001] DPT_Switch
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					
444	[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.					
445	[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					
445	[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					
446	[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.					
447	[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					
448	[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.					

3 – Parameter page: General Settings

Parameter	Settings
DEVICE NAME	**SA04K01KNX**
Here a personalized name for each device can be entered. E.g. **SA04K01KNX living room**	
Channes A, B, C, D	No **Shutter / Blind**
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 SA04K01KNX 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**
Each cannel can be configured 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.	
Central ON/OFF, UP/DOWN object	**No** Yes
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 Central UP/DOWN objects for shutter/blind.	
Manual control	Param Mode + Test Mode Param Mode Test Mode Disable
The SA04K01KNX 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 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.	

4.1.1 – 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 SA04K01KNX 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 Attention! Manual control must be activated in outputs	No **Yes**
The SA04K01KNX 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.	

4.1.2 – Bus failure

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

Down: whenever the bus voltage returns, the channel moves DOWN. The first relay will be opened; and the second relay will be closed for the full "Travel time movement UP", independent of the current blind position. If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN.
Position: whenever the bus voltage returns, the shutter will move to a certain position (0-100%), which can be parameterized here.
Move to slat and blind position: not applicable for shutter configuration.
Blinds (with slats): whenever the bus voltage returns, the blind and the slats will move to a certain position (0-100%)
Preset: you can select one of the four previously configured PRESETS (Channel/Advanced Functions) to be executed on bus voltage recovery.

Attention! Presets parameters must be configured in Channel -> Advanced functions
Recovery status before bus failure: the status of the output will be saved in the actuator's non-volatile memory; therefore, when the actuator initializes, if this option has been chosen, it will move the shutter to the position previous to the bus failure.

Important note on calibration: for "Position", "Move to slat and blind position", "Preset" and "Recovery status before bus failure".

Attention! An absolute position on bus power recovery will cause a calibration movement to the upper end position
Sometimes it is impossible for the actuator to know the exact position of the shutter: for instance, on bus voltage return (the power failure of the bus and that of the current shutter are independent from each other) or with heavy shutters having made several absolute position movements (without having reached the end position).

In these cases, the actuator needs to calibrate itself by making a full movement to the 0/100% position (upper/lower end position) before moving to the desired absolute position.

After calibration, the shutter now has a reference from where to part again for the next movement.

4.1.3 – 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 SA04K01KNX 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.	

4.1.3.1 – 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
his 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.	

Additional waiting time delays

Parameter	Settings
Additional waiting time delays	**No** Yes
The function “Additional waiting time delays” can be used when the actual shutter/blind up/down movement is delayed after the relays are switched. It occurs due to the inertia of some motors or mechanical reasons of the blind.	
Time for blind to start moving after UP movement	**No** Yes
Base Factor	**10ms** 1
This time will only be applied when the blind is completely lowered with the slats or slits closed (Blind & Slat = 100%). When an UP movement action is received, the parameterized time is added to the total drive time, in order to take into account the time the motor needs to start moving the blind upwards. This way the calculation of the blind position is accurate.	
Delay on any movement	**No** Yes
This time is used when the blind is in any position. On any movement, the parameterized time is added to the total drive time, in order to take into account the time the motor needs to start moving.	
Deceleration delay	**No** Yes
Base Factor	**10ms** 1
This time is used to compensate for the inertia of the blind to stop moving. Therefore, the actuator channel will calculate the exact time to open the relay, taking into account the deceleration extra time which the motor will need to stop moving in order to reach the desired position.	

Delay when changing Slat direction from DOWN to UP	**No** Yes
Base Factor	**10ms** 1
This time is used to compensate the time the slats take to change direction. It is the time it takes to tighten the slats strings until it starts to move in opposite direction. This time will only be applied on an Slat UP movement and if the previous action was a DOWN movement.	
Delay when changing Slat direction from UP to DOWN	**No** Yes
This time is used to compensate the time the slats take to change direction. It is the time it takes to tighten the slats strings until it starts to move in opposite direction. This time will only be applied on an Slat DOWN movement and if the previous action was a UP movement.	

4.1.3.2 – 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.	

4.1.3.3 – 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.	

4.1.3.4 – 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:

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

Nothing: the channel will not participate in the alarm. Thus, it will not be blocked.
Block channel as is: the channel will be blocked, but not move when activating the alarm. Should the alarm be triggered while the blind is moving, the blind will stop immediately and the current status will be sent to the bus.
Move Up: the channel moves UP. The second relay will be opened; and the first relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position)
Move Down: the channel moves DOWN. The first relay will be opened; and the second relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position). If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN, and thus the remaining time will be calculated accordingly.
Move to position: the shutter will move to a certain position (0-100%) when executing the alarm:
Only movement position: the exact position can be parameterized:
Only slat position: not applicable for shutter configuration.
Blinds (with slats): the exact position of the slats can be parameterized here.
Movement and slat position: not applicable for shutter configuration.
Blinds (with slats): the exact position of the blind and of the slats can be parameterized:
Move to preset: you can select one of the four previously configured PRESETS (Channel/Advanced Functions) to be executed on alarm.

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

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

Nothing: the channel will not do anything at the end of all alarms.
Move Up: the channel moves UP. The second relay will be opened; and the first relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position)
Move Down: the channel moves DOWN. The first relay will be opened; and the second relay will be closed during the remaining time (since the actuator knows the complete TRAVEL TIME MOVEMENT UP, it will now calculate the travel time still needed to complete the full movement depending on the current position). If a different time has been defined for moving down, then the time for a full movement will be the DIFFERENT TRAVEL TIME FOR MOVEMENT DOWN, and thus the remaining time will be calculated accordingly.
Move to position: the shutter will move to a certain position (0-100%) at the end of all alarms.
Only movement position: the exact position can be parameterized:
Only slat position: not applicable for shutter configuration.Blinds (with slats): the exact position of the slats can be parameterized.
Movement and slat position: not applicable for shutter configuration.
Blinds (with slats): the exact position of the blind and of the slats can be parameterized.

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

Even though the actuator does not move, it does register all the absolute position events (not the one bit movements, like up/down, slat up/down) in order to be able to go to the state where it would have been at enabling (if the channel had not been blocked).
Attention! *The “Behaviour at the end of all alarms” will only be executed with no active & acknowledged channel alarms, and if the “disable channel function” is in enabled state. Only then, the channel will be unblocked.*

4.1.3.5 – 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.*

4.1.4 – 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: 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”.

5.1 - 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.	

5.1.1 – 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.	

5.1.2 – 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.	

5.1.3 – 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.	

5.2 – Logics

There are 25 logic functions available in SA04K01KNX o16 and 35 in SA04K01KNX 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.	

5.2.1 – 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.	

5.2.1.1 – 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.	

5.2.1.2 – 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	

5.2.2 – 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.	

5.2.2.1 – 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.	

5.2.2.2 – 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	

5.2.3 – 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.	

5.2.3.1 – 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.	

5.2.3.2 – 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	

5.2.4 – 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.	

5.2.4.1 – 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.	

5.2.4.2 – 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	

5.2.5 – 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.	

5.2.5.1 – 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.	

5.2.5.2 – 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	

5.3 – 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	

5.3.1 – 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.	

5.4 – 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.	

5.4.1 – 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.	

5.4.1.1 – 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.	

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.

5.4.1.2 – 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.	

5.5 – 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)	

5.5.1 – Setpoint 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.	

5.5.2 – 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"	

5.5.2.1 – 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.	

5.5.3 – 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"	

5.5.3.1 – 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.	

5.6 – 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.	

5.6.1 – 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.	

5.7 – 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	

5.7.1 – 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	

5.7.1.1 – Input object

Parameter	Settings
Output object to send variable	General 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	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.	

5.7.1.2 – Output object

Parameter	Settings
Input object to send variable	General **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 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	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.	

5.8 – 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.	

5.9 – End-user 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.	

5.9.1 – 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 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	

5.9.1.1 – End-user 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	

5.10 – 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.	

5.11 – 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.	

6 – 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**.

7 – Annex

7.1 – 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 Shutter/Blinds.
The LEDs represent:
For Shutter/blinds: The top row: channel’s first relay A->UP, A->DOWN, B-UP, etc.

7.1.1 – Parameter Mode

Manual Control – Parameter Mode The Parameter Mode allows you to control all the channels of the actuator as configured in the ETS. The Action simulates a telegram received at the switching object of the selected channel.
SHUTTER/BLIND
Long press action (Channel output 1): Sends an UP command “0” to the “Move” object. Long press action (Channel output 2): Sends a DOWN command “1” to the “Move” object. Short press action (any output) (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

7.1.2 – 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.

To change into the test mode, any button can be used depending of the channel configuration:
- 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:

SHUTTER/BLIND
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)

Eelectron SpA
Via Monteverdi 6,
I-20025 Legnano MI, Italia

Tel: +39 0331.500802
Fax: +39 0331.564826
E-mail: info@eelectron.com
Web: www.eelectron.com