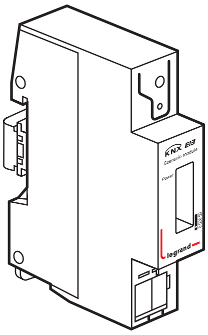
KNX- scenario and events module

Catalogue number(s): 0 026 50



CONTENTS Page

1. USE

The KNX scenarios/events module Cat. No. 0 026 50 is a device installable on a DIN rail . The scenarios/events module application program includes control functions to set and program events that can be selected during the configuration of the parameters.

Scenarios control
A scenario is a set of statuses or defined values that are sent to many different actuators. This can be used, per instance, to control with a single command the lights , blinds and screens of a classroom during a presentation.

Main functions :
- Programming and recalling of 8 scenarios involving up to 8 different actuations
- Management of data types from 1 bit to 14 bytes
- Shared or distinguished communication objects for each group with regard to the different scenarios
- Saving and recalling of the scenarios through activation commands through 1-bit or 8-bit telegrams
- Easy conversion of 1 bit in 8 bits scenario commands and vice versa
- Scenarios association to values from 1 to 64 in the management of scenarios through 8-bit telegrams
- Variable time delay to recall the scenarios configurable between 0 and 30 minutes

Events programming
The function of event programming allows to define and recall events. The recalling of an of event triggers the sending of multiple telegrams to the KNX BUS in order to enable multiple functions and configure multiple parameters simultaneously.

Main functions:
- Programing and recalling up to 8 different events each of them made by a maximum of 10 activities
- Management of data types from 1 bit to 14 bytes
- Recalling of events through 1-bit to 8 -byte telegrams
- Possibility to trigger events or conditions on the basis of defined reference values
- Possibility to lock/unlock of events programming
- Variable time delay to recall the event configurable between 0 and 120 minutes

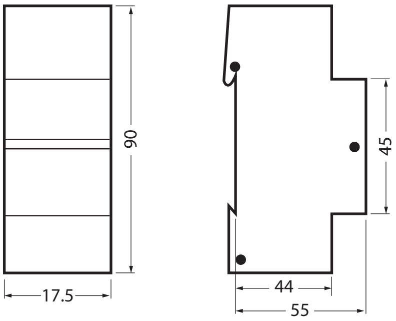
2. TECHNICAL FEATURES

2.1 Climatic features
- Ambient operating temperature: - 5 to + 45°C
- Storage temperature: - 25 to + 70°C
- Relative humidity (non-condensing): 5 to 93%

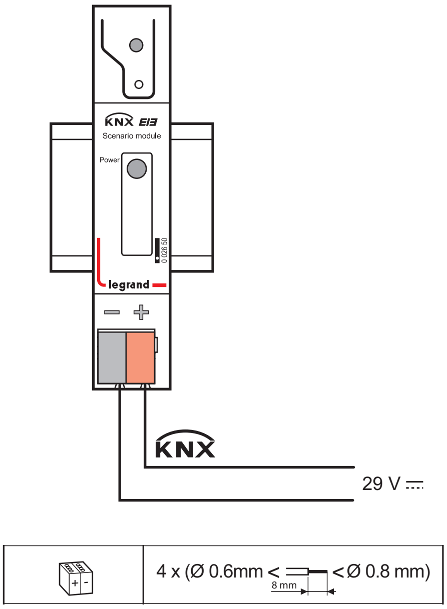
2.2 Electrical features
- KNX/BUS power supply: 29 V⎓
- KNX/BUS absorption: 6 mA
- Automatic clamps

2.3 Mechanical features
- Protection class (in accordance with standard EN 60529): IP 20
- Weight: 65g

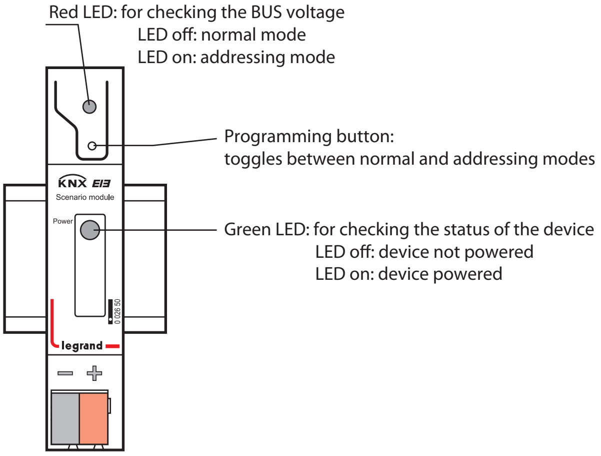
3. OVERALL DIMENSIONS



4. CONNECTION



5. OPERATION



Red LED: for checking the BUS voltage
LED off: normal mode
LED on: addressing mode

Programming button:
toggles between normal and addressing modes

Green LED: for checking the status of the device
LED off: device not powered
LED on: device powered

Scenario control
The “scenario control” function can program and call up a maximum of 8 scenarios. States (for example, lighting on/off, or venetian blinds raised/lowered position) or values (for example, lighting at 70%, or temperature at 22°C) can be controlled for each scenario.

The default values of the states required for each scenario are configured using ETS 3 or 4 software. The user can also modify scenarios when they are started, program them using the scenario module and call them up at a later date.

When a scenario is under programming, the actuators involved in this scene are first of all set to the required state (for example, lighting at 70%, or venetian blinds lowered, etc.) using the control devices provided. The scenario module queries the states of the actuators and then assigns them to the corresponding scenario number.

5. OPERATION (CONTINUED)

Scenario control (continued)
The programmed values are retained, even after the scene/event controller has been restarted (for example, after a default in the BUS current).

A variable delay of between 0 and 30 minutes can be applied to the transmission of the various telegrams when a scene is called up.

The following types of communication object data can be handled:
- Activation (1 bit)
- Venetian blinds (1 bit)
- Forced control (2 bits)
- Value as a percentage (1 byte)
- Counter value (1 byte)
- ASCII characters (1 byte)
- Floating point value (2 bytes)
- Counter value (2 bytes)
- Floating point value (4 bytes), counter value (4 bytes)
- Information on errors (6 bytes)
- Character string (14 bytes)
- Scenes (1 bit and 8 bits)

Scenes are called up and programmed using 1-bit activation commands or 8-bit scene telegrams. It is also possible to define various groups of scenes in the form of scene calls (1 bit) or scene telegrams (8 bits). This simplifies the conversion of 1-bit scene openings to 8 bits and vice versa.

It is possible to use additional scene/event modules to increase the number of scenes or group addresses in a scene.

Event programs
The application program can develop up to 8 event programs, and it is possible to trigger up to 10 event tasks, with various types of data, within each program.

For example:
- Activation of venetian blinds/scene (1 bit)
- Forced control (2 bits)
- Relative dimming (4 bits)
- Value as a percentage (1 byte)
- Counter value (1 byte)
- ASCII characters (1 byte)
- Call up scene (8 bits)
- Scene programme (8 bits)
- Floating point values (2 and 4 bytes)
- Light level and wind speed type values
- Counter values (2 and 4 bytes)
- Operating times, information on errors (6 bytes)
- Character strings (14 bytes)

The default levels of the values required for each scene are defined when the parameters are configured with the ETS 3 and 4 software.

A variable delay of between 0 and 120 minutes can be applied to the transmission of the event telegrams when an event program is called up.

It is possible to use BUS telegrams with various data types, such as 1-bit values, 2-bit forced control, 4-bit values, 8-bit values, 2 and 4-bytes counter and floating point values, 3-bytes time values, 6-bytes information on errors or 8-bytes system time information as trigger events. The event programs are opened using reference values and opening conditions. The event programmes can also be unlocked and locked.

Configuration
It is necessary to use ETS 3 or 4 software to assign parameters and address the scene/event controller Cat. No. 0 026 50.

6. STANDARDS AND APPROVALS

Electrical safety

- Degree of pollution (in accordance with standard IEC 60664-1): 2
- Protection class (in accordance with standard IEC 61140): III
- Overvoltage class (in accordance with standard IEC 60664-1): III
- BUS: safety extra low voltage (SELV) 24 VDC
- The device complies with standards EN 50090-2-2 and IEC 60664-1
- Complies with standards EN 61000-6-2, EN 61000-6-3 and EN 50090-2-2

Marking

- KNX EIB, CE

Note: All technical information is available at

www.legrandoc.com

7. MAINTENANCE

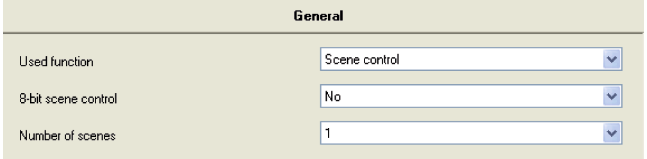
Clean the surface with a cloth.
Do not use acetone, tar-removing cleaning agents or trichloroethylene.

Caution: Always test before using other special cleaning products.

8. COMMUNICATION OBJECTS

Use the "General" parameters page to select the desired scenario module function

General



General

Used function: Scene control

8-bit scene control: No

Number of scenes: 1

Parameter	Settings
Used Function	Scenes control Event program
This parameter determines whether the scenes control or event program is to be used. **Note:** Subsequent changes to this parameter leads to losing the group addresses in the relevant objects.	

8.1 Communication objects scenes control

The following communication objects will be displayed in the scenario module if scene control has been selected in the "General" parameters window and all functions have been enabled.

Obj	Object name	Function	Type	Flag
1	Scene 1 group A send/recall	Object	1 bit	Send CRTU
1	Scene 1 group A send/recall	Object	2 bits	Send CRTU
1	Scene 1 group A send/recall	Object	1 byte	Send CRTU
1	Scene 1 group A send/recall	Object	2 bytes	Send CRTU
1	Scene 1 group A send/recall	Object	4 bytes	Send CRTU
1	Scene 1 group A send/recall	Object	6 bytes	Send CRTU
1	Scene 1 group A send/recall	Object	14 bytes	Send CRTU

8. COMMUNICATION OBJECTS (CONTINUED)

On calling up scene 1, this object sends the value programmed for this purpose to group A. You use the "data type" parameter for this scene and group to select the desired object type. If the setting is "not used" this object is not displayed.
According to the data type chosen, this object is linked via the appropriate group address with the corresponding actuator object (e.g. switching, Venetian blind up/down, forced control, brightness value in %, floating point value, count value, character string) to the device to be controlled with this group.
The currently programmed Scene 1, Group A value can be read, to control, through this object.

Note:
The "14 bytes character string" is always present only in the relevant group A of scenes 1 to 8.
The "6 bytes fault information string" is always present only in the relevant group B of scenes 1 to 8.

Obj	Object name	Function	Type	Flag
2	Scene 1 group A read/program	Object	1 bit	Read CWTU
2	Scene 1 group A read/program	Object	2 bits	Read CWTU
2	Scene 1 group A read/program	Object	1 byte	Read CWTU
2	Scene 1 group A read/program	Object	2 bytes	Read CWTU
2	Scene 1 group A read/program	Object	4 bytes	Read CWTU
2	Scene 1 group A read/program	Object	6 bytes	Read CWTU
2	Scene 1 group A read/program	Object	14 bytes	Read CWTU

About this object the current value of the group of A is questioned when programming scene 1. The data type is identical to the setting made for the "Scene 1 Group A" object. If this is set to "No" when choosing "object read/program use", this object is not displayed.

According to the data type chosen, this object is linked via the appropriate group address with the corresponding status object (e.g. switching, Venetian blind up/down, forced control, brightness value in %, floating point value, count value, character string) to the device to be read. To read the current value or status and program it into the scenes-/event controller, the group address is to be entered as "sending address" in the object of the device to be interrogated and the read flag has to be set.

Note:
The "14 bytes character string" is always present only in the relevant group A of scenes 1 to 8.

The "6 bytes fault information string" is always present only in the relevant group B of scenes 1 to 8.

The communication objects 3 - 16 have the same meaning for Scene 1, Groups B to H.

The communication objects 17 - 128 have the same meaning for Scenes 2 to 8, Groups A to H. These objects are only present if proprietary objects are used conditionally for calling up and programming through the parameter setting for the corresponding scenes.

Obj	Object name	Function	Type	Flag
129	Scenes 1/2 recover	recall	1 bit	Received CWTU
This object calls up Scenes 1 and 2. If a "0" telegram is received, the scenes-/event controller sends the programmed scene 1 values via the "send/recall group" objects to the addressed actuators. Scene 2 is called up correspondingly on receiving a "1" telegram.				

8. COMMUNICATION OBJECTS (CONTINUED)

8.1 Communication objects scenes control (continued)

The communication objects 130 - 132 have the corresponding meaning for scenes 3/4, 5/6 and 7/8.

Obj	Object name	Function	Type	Flag
133	Scenes 1/2 program	program	1 bit	Received CWTU
This object programs Scenes 1 and 2. If a "0" telegram is received, the scenes-/event controller reads out the currently set statuses or values of the corresponding actuators via the "read/program group" objects and programs these in scene 1. Scene 2 is programmed correspondingly on receiving a "1" telegram.				

The communication objects 134 - 136 have the corresponding meaning for scenes 3/4, 5/6 and 7/8.

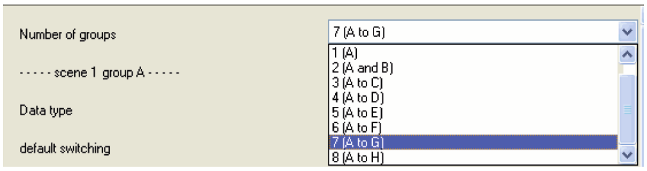
Obj	Object name	Function	Type	Flag
137	8-bit scene	read/ program	1 byte	Received CWTU
This object calls up or programs the scene for which a telegram with the value (1-64) is received, which is set for this scene in the parameter "Scene number at 8-bit scene control".				

Scene control parameters

General

Parameter	Settings
8-bit scene control	No Yes
You use this parameter to set whether the scenes are also to be called up and programmed via 8-bit scene openers. If so, the communication object "8-bit scene" will be created, as well as a parameter for selecting the scene number, with which this is addressed for each scene.	
Number of scenes	1 2 3 4 5 6 7 8
This parameter determines the number of scenes which are to be used in the scenes-/event controller. This selection will fade in the parameter pages for setting and the communication objects for calling up and programming approved scenes.	

Scene 1 Group A...



Number of groups: 7 (A to G)

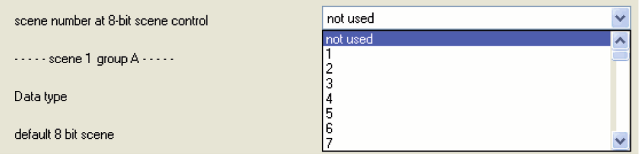
1 (A)
2 (A and B)
3 (A to C)
4 (A to D)
5 (A to E)
6 (A to F)
7 (A to G)
8 (A to H)

- - - - - scene 1 group A - - - - -

Data type

default switching

Parameter	Settings
Number of groups	none 1 (A) 2 (A and B) 3 (A to C) 4 (A to D) 5 (A to E) 6 (A to F) 7 (A to G) 8 (A to H)
This parameter determines the number of groups used in scene 1. This selection creates additional parameters to initialize default settings and communication objects for the groups used. If you set "none", scene 1 is not used.	
Scene number at 8-bit scene control	not used 1 2 64
You use this parameter to set whether 8-bit scene control is used for scene 1 and with which scene number (1-64) it is called up and programmed. **Note:** All scenes to be called up must be called up or programmed under different scene numbers. If several scenes are set with the same scene number for calling up or programming scenes, only the first program will be called up or programmed.	



scene number at 8-bit scene control: not used

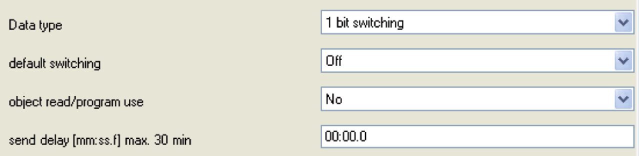
not used
1
2
3
4
5
6
7

- - - - - scene 1 group A - - - - -

Data type

default 8 bit scene

Scene 1 Group A...



Data type: 1 bit switching

default switching: Off

object read/program use: No

send delay [mm:ss.f] max. 30 min: 00:00.0

8. COMMUNICATION OBJECTS (CONTINUED)

8.1 Communication objects scenes control (continued)

Parameter	Settings
Data type	not used 1 bit switching 1 bit Venetian blind 1 bit scene 2 bit forced control 8-bit scene 1 byte % value 1 byte count value 1 byte ASCII character 2 byte floating point 2 byte count value 4 byte floating point 4 byte count value 6 byte fault information 14 byte character string
This parameter determines the data type for the "send/recall up Group A" and "read/program Group A" communication objects. If the setting is "not used", this group is not used in scene 1 and the corresponding parameters and communication objects are faded out. This is how you can delete or add individual groups in different scenes. **Note:** The "14 bytes character string" is always present only in the relevant group A of scenes 1 to 8. The "Fault information" data type is always present only in the relevant group B of scenes 1 to 8. On BUS power recovery "send/recall up Group A" communication objects are set to zero.	

Different value ranges are allowed for the following parameters, depending on the group data type set.

Scene 1 Group A…

Parameter	Settings
default switching	Off On
default venetian blind	Up Down
default 1-bit scene	Scene A Scene B
default forced control	Forced control Off (0) Force controlled to Off (2) Force controlled to On (3)
default 8-bit scene	1 2 … … 64
default value in %	0
default 1-byte count value	0
default ASCII characters	0
default 2-byte floating point	0
default 2-byte count value	0
default 4-byte floating point	0
default 4-byte count value	0
default 6-byte fault information	0x00:0:00:00:00:00
default 14-byte character string	0

This parameter sets the initial Group A default value for scene 1. This is how you plan scene settings and use these directly after starting up the scene-/event controller. The value is held in the controller until it is overwritten when programming scenes by reading out the current status of the linked actuator.
If the communication object "recall/program Group A" is not used, or is not populated with a group address, which can be interrogated via the BUS, the preset value here is maintained when programming scenes. The value is maintained even after a BUS power outage.

Note:
Entries are limited to counts in the maximum possible value range in each case:

Value in %	0 to 100
1-byte count value	0 to 255
1-byte ASCII character	Entry (a character)
2-byte floating point	-671,088.64 to 671,088.96
2-byte count value	0 to 65,535
4-byte floating point	-3.40 E38 to 3.40 E38
4-byte count value	0 to 4,294,967,295
6-byte fault information	to KNX specification, DPT_Alarminfo
14-byte character string	Text input, max. 14 characters

Scene A (corresponds to scenes 1, 3, 5 and 7 and is called up via the value "0".
Scene B (corresponds to scenes 2, 4, 6 and 8 and is called up via the value "1".

Note:
If a value is sent to the object "read/program Group A" (e.g. by using the ETS, an external display-/control unit or a visualisation) and scene 1 is then programmed, without the temporary value of the object having been overwritten (because the object "read/program Group A" is not linked with the status object of the actuator), this value will be programmed as the new scene value and sent in future when the scene is called up. This is how you can program a new scene value without moving the actuator to this status in advance and reading it out to program the scene.

Scene 1 Group A…

Parameter	Settings
object read/program use	No Yes
You use this parameter to set whether the object "read/program Group A" is to be created. If so, the communication object is created. You use this communication object to read and program the current state of the status object, the actuator, in each case. **Note:** When using the "8-bits scene" data type, the parameter setting "Use read/program object" does not pop up. In an 8-bit scene, you call up and program a scene with the "read/program" communication object. The highest value bit 7 states whether the scene is called up (bit = 0) or programmed (bit = 1). On BUS power recovery, "read/program object" is set to zero.	

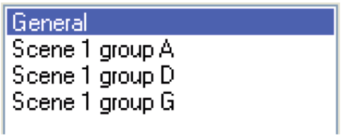
8. COMMUNICATION OBJECTS (CONTINUED)

8.1 Communication objects scenes control (continued)

Scene 1 Group A…

Parameter	Settings
send delay [mm:ss.f] max. 30 min	00:00.0
This parameter determines now much time should elapse between the scene opening and sending of the scene 1 group A value. This is how you can comply with a given sequence when sending group telegrams on calling up a scene. This is how you can produce dynamic functions, such as staircase lighting control with preliminary warning, by linking an actuator with various groups in a scene. If a scene is called up again, then processing is suspended before all values have been sent owing to the time delay. All time delays will be reset and restarted. **Note:** The time input is limited to a value range between 0.1 seconds and 30 minutes.	

Depending on the number of groups to be used, the settings for these groups are displayed in a maximum of 3 windows, Scene 1 Group A…, Scene 1 Group D… and Scene 1 Group G…


General
Scene 1 group A
Scene 1 group D
Scene 1 group G

Scene 1 Group A…

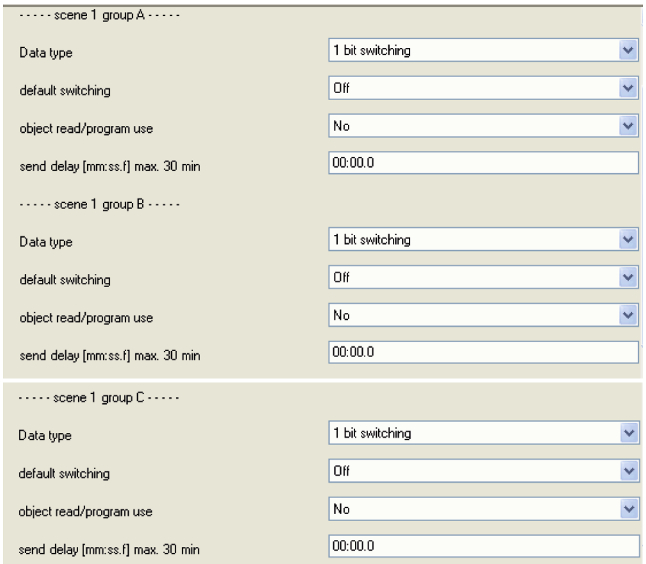
You use Scene 1 Group A… to select settings up to Group C.

Scene 1 Group D…

You use Scene 1 Group D… to select settings up to Group F.

Scene 1 Group G…

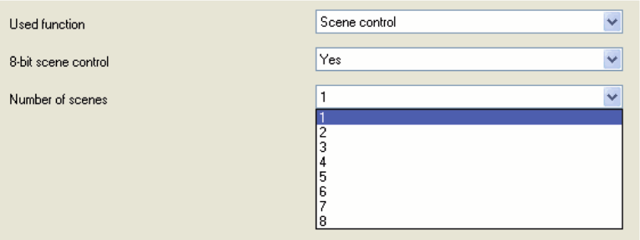
You use Scene 1 Group G… to select settings up to Group H.


- - - - - scene 1 group A - - - - -
Data type: 1 bit switching
default switching: Off
object read/program use: No
send delay [mm:ss.f] max. 30 min: 00:00.0
- - - - - scene 1 group B - - - - -
Data type: 1 bit switching
default switching: Off
object read/program use: No
send delay [mm:ss.f] max. 30 min: 00:00.0
- - - - - scene 1 group C - - - - -
Data type: 1 bit switching
default switching: Off
object read/program use: No
send delay [mm:ss.f] max. 30 min: 00:00.0

The parameters for groups B to H are identical to those for Group A. They will only be displayed if they have been enabled via the "Number of groups" parameter.

Note:
The "14 bytes character string" is always present only in the relevant group A of scenes 1 to 8.
The "Fault information" data type is always present only in the relevant group B of scenes 1 to 8.

Scenes 1…8

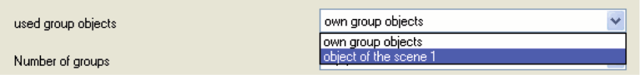

Used function: Scene control
8-bit scene control: Yes
Number of scenes: 1

You use the parameter "Number of scenes" to set how many scenes are to be used. The communication objects and parameters will be visible according to the number of scenes set. The parameters for scenes 1 and 5, for the relevant groups, are identical.

To reduce the setting overheads, there is an option, to be determined for each enabled scene, whether proprietary sending objects or sending objects for the first allocated scene range are to be used.
For this purpose, 8 scenes are divided into 2 scene ranges. Each scene range can be allocated a maximum of 4 scenes. You can allocate scenes 2, 3 and 4 to scene 1. Scene 1 is therefore the first scene range. You can allocate scenes 6, 7 and 8 to scene 5.
Scene 5 is therefore the first scene range.

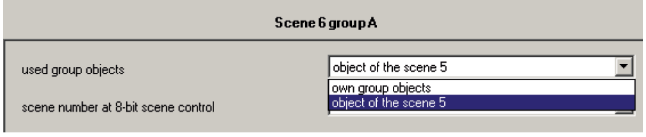
Note:
For all scenes which are allocated to a scene range, only one "send/read object" and one "read/program object" are available and these must be linked with a group address.
If proprietary sending objects are used for individual scenes, combinations are possible in which, in the various scenes, individual actuators can be dispensed with or added, or a scene number can be disabled completely. In 1-bit scene opening, 2 scenes are always combined and are therefore always present in pairs.

Accordingly, scenes 2, 3 and 4 have an additional parameter:


used group objects: own group objects
Number of groups

Parameter	Settings
used group objects	own group objects object of the scene 1
This parameter determines whether proprietary sending objects are to be provided for the selected groups in each case or whether scene 1 sending objects are to be used for scenes 2, 3 and 4.	

Accordingly, scenes 6, 7 and 8 have an additional parameter:


Scene 6 group A
used group objects: object of the scene 5
scene number at 8-bit scene control

Parameter	Settings
used group objects	own group objects object of the scene 5
This parameter determines whether proprietary sending objects are to be provided for the selected groups in each case or whether scene 5 sending objects are to be used for scenes 6, 7 and 8.	

General planning note:
In theory, a large number of random channels can be addressed via a single scene by using scenes with the same opener in different controllers. You should therefore note that, at the same time, the BUS load through sending individual scene values does not become too great, because otherwise the BUS would be blocked temporarily.

8. COMMUNICATION OBJECTS (CONTINUED)

8.2 Event program communication objects

The following communication objects will be displayed in the 0 026 50 scene-/event controller if event program has been selected in the "General" parameters window and all functions have been enabled.

Obj	Object name	Function	Type	Flag
1	EP 1: event opening	Object	1 bit	Received CWTU
1	EP 1: event opening	Object	2 bits	Received CWTU
1	EP 1: event opening	Object	4 bits	Received CWTU
1	EP 1: event opening	Object	8 bits	Received CWTU
1	EP 1: event opening	Object	1 byte	Received CWTU
1	EP 1: event opening	Object	2 bytes	Received CWTU
1	EP 1: event opening	Object	3 bytes	Received CWTU
1	EP 1: event opening	Object	4 bytes	Received CWTU
1	EP 1: event opening	Object	6 bytes	Received CWTU
1	EP 1: event opening	Object	8 bytes	Received CWTU

Use this object to call up event program 1. On receiving a telegram with a value for event opening or on receiving a telegram which fulfils the condition when compared with the currently valid comparator value and the event program is approved, the controller sends the set event task values for event program 1 to the addressed actuators via the objects "Event task A" to "Event task K".
You use the "Event opening / reference value data type" parameter to select the desired object type.

Note:
The difference between calling up an 8-bit scene and programming an 8-bit scene is that when calling up an 8-bit scene the highest value bit is set to 0, whereas when programming an 8-bit scene the highest value bit is set to 1. This fact should be borne in mind when comparing the event opening with the comparator value.

Obj	Object name	Function	Type	Flag
2	EP 1: reference value	Object	1 bit	Received CWTU
2	EP 1: reference value	Object	2 bits	Received CWTU
2	EP 1: reference value	Object	4 bits	Received CWTU
2	EP 1: reference value	Object	8 bits	Received CWTU
2	EP 1: reference value	Object	1 byte	Received CWTU
2	EP 1: reference value	Object	2 bytes	Received CWTU
2	EP 1: reference value	Object	3 bytes	Received CWTU
2	EP 1: reference value	Object	4 bytes	Received CWTU
2	EP 1: reference value	Object	6 bytes	Received CWTU
2	EP 1: reference value	Object	8 bytes	Received CWTU

The reference value is received via this object and it is then compared with the event opening value to decide whether the event program is to be called up. If no value has been received, the set reference value is used.
You use the "Event opening / reference value data type" parameter to select the desired object type.
A received reference value is programmed if the voltage fails and after the power recovery, the voltage is preset for a fresh comparison.
The currently set comparator value can be read out for checking. The read flag must be set for this purpose.

Note:
If the object is interrogated actively after the ETS download, then the interrogated value will not equal the currently applicable value.
This is only valid after the object is updated.

The communication objects:
17 = EP 2: Event opening
33 = EP 3: Event opening
49 = EP 4: Event opening
65 = EP 5: Event opening
81 = EP 6: Event opening
97 = EP 7: Event opening
113 = EP 8: Event opening
and have the same meaning as in EP 1: Event opening.

The communication objects:
18 = EP 2: Reference value
34 = EP 3: Reference value
50 = EP 4: Reference value
66 = EP 5: Reference value
82 = EP 6: Reference value
98 = EP 7: Reference value
114 = EP 8: Reference value
and have the same meaning as in EP 1: Reference value.

8. COMMUNICATION OBJECTS (CONTINUED)

8.2 Event program communication objects (continued)

Obj	Object name	Function	Type	Flag
3	EP 1: release/locking	Object	1 bit	Received CWTU
Use this object to release or locking the event program. If the event program is locking, the event telegrams A to K are not sent receiving an event opening telegram, even if the comparison is positive. The release or locking EP status is programmed if the voltage fails and is preset after the power recovery. **Note:** If the object is queried after the ETS download, then the interrogated value will not be equal to the currently applicable value. This is only valid after the object is updated. The read flag must be set for this purpose.				

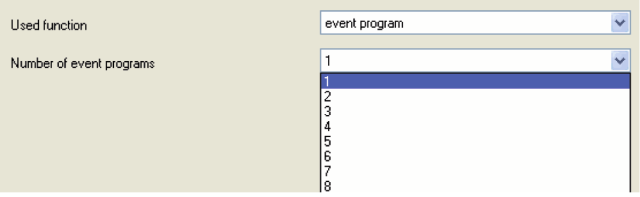
The communication objects:
19 = EP 2: Release/Locking
35 = EP 3: Release/Locking
51 = EP 4: Release/Locking
67 = EP 5: Release/Locking
83 = EP 6: Release/Locking
99 = EP 7: Release/Locking
115 = EP 8: Release/Locking
and have the same meaning as in EP 1: Release/Locking.

Obj	Object name	Function	Type	Flag
4	EP 1: event task A	Object	1 bit	Send CTU
4	EP 1: event task A	Object	2 bits	Send CTU
4	EP 1: event task A	Object	4 bits	Send CTU
4	EP 1: event task A	Object	1 byte	Send CTU
4	EP 1: event task A	Object	8 bits	Send CTU
4	EP 1: event task A	Object	2 bytes	Send CTU
4	EP 1: event task A	Object	4 bytes	Send CTU
4	EP 1: event task A	Object	6 bytes	Send CTU
4	EP 1: event task A	Object	14 bytes	Send CTU
Use this object to send the event task A value when the event program opens. You use the "Data type" parameter for this event task to select the desired object type. Depending on the data type selected, this object is to be linked via group address with the corresponding actuator object of the device to be controlled with this event task. **Note:** The "14 bytes character string" is always present only in the relevant event task A of event programs 1 to 8. The "6 bytes character string" is always present only in the relevant event task B of event programs 1 to 8.				

The communication objects:
5-13 = EP 1: Event tasks B-K
20-29 = EP 2: Event tasks A-K
36-45 = EP 3: Event tasks A-K
52-61 = EP 4: Event tasks A-K
68-77 = EP 5: Event tasks A-K
84-93 = EP 6: Event tasks A-K
100-109 = EP 7: Event tasks A-K
116-125 = EP 8: Event tasks A-K
and have the same meaning as in EP 1: Event task A. These objects are only present if proprietary objects are used to send the corresponding event tasks if conditioned by the parameter settings.

Event program parameters

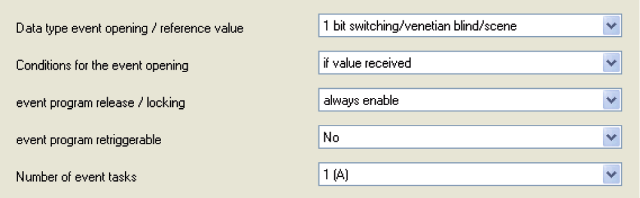
General



Used function: event program
Number of event programs: 1

Parameter	Settings
Number of event programs	1 2 3 4 5 6 7 8
This parameter determines the number of event programs which are to be used in the scene-/event controller. Its selection fades in the parameter pages for setting and the communication objects for the corresponding event programs.	

Event program 1



Data type event opening / reference value: 1 bit switching/venetian blind/scene
Conditions for the event opening: if value received
event program release / locking: always enable
event program retriggerable: No
Number of event tasks: 1 (A)

8. COMMUNICATION OBJECTS (CONTINUED)

8.2 Event program communication objects (continued)

Parameter	Settings
Data type event opening / reference value	1 bit switching/ venetian blind/scene 2 bit forced control 4 bit dimming 8 bit scenes recall 8 bit scenes program 1 byte value % 1-byte count value 2 byte floating point 2 byte count value 3 byte time 4 byte floating point 4 byte count value 6 byte fault information 8 byte system time
This parameter determines the data types for the communication objects "EP 1: Event opening" and "EP 1: Reference value". There is accordingly the possibility of using any BUS telegrams with different values to open event programs.	

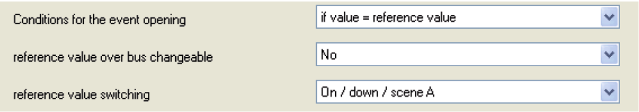
Event program 1

Parameter	Settings

Condition for the event opening	if value received
	if value = reference value
	if value > reference value
	if value < reference value
	if reference value is exceeded
	if reference value is undershot

This parameter defines the condition with which the event program is to be called up. The value of a telegram received on the object "EP 1: Event opening" will be compared with the current reference value according to the parameter setting. If the condition is fulfilled and the EP is enabled, this will be run. **Note:** If the 6-byte fault information data type is used, the settings "if reference value is exceeded" and "if reference value is undershot" are not available.	

Event program 1



Conditions for the event opening	if value = reference value
reference value over bus changeable	No
reference value switching	On / down / scene A

Parameter	Settings
reference value over BUS changeable	No Yes
This parameter defines whether the set reference value can be changed during operation via the object "EP 1: Reference value".	

If the setting is "No", this object is not available.
A received reference value is programmed if the voltage fails and after the power recovery, the voltage is preset for a fresh comparison.

Note:
There is the possibility to change the reference value during operation via an external display-/control unit or via the BUS.

Event program 1

Parameter	Settings
reference value switching	On/down/scene A Off/up/scene B
reference value forced control	Forced control Off (0) Force controlled to Off (2) Force controlled to On (3)
reference value dimming relative	Stop (0) 100% darker (1) ½ darker (2) 1/4 darker (4) 1/8 darker (4) 1/16 darker (5) 1/32 darker (6) 1/64 darker (7) 100% brighter (9) ½ brighter (10) 1/4 brighter (11) 1/8 brighter (12) 1/16 brighter (13) 1/32 brighter (14) 1/64 brighter (15)
reference value scene number	1, 2, …, 64
reference value in %	0
reference value count value	0
reference value floating point	0
reference value time	00:00:00
reference value fault information AlarmPriority: ApplicationArea: Error- Class	0x0:00:00
Reference value system time hh:mm:ss	00:00:00

Use this parameter to initialise the default reference value with which a value received via the object "EP 1: Event opening" will be compared. The reference value data type is the same as the event opening data type. If the condition is fulfilled and the EP is enabled, this will be run. The set reference value can be changed via the object "EP 1: reference value".

Note:
Entries are limited to counts in the maximum possible value range in each case:

Value in %	0 to 100
1 byte count value	0 to 255
2 byte floating point	-671,088.64 to 671,088.96
2 byte count value	0 to 65,535
3 byte time	00:00:00 to 23:59:59
4 byte floating point	-3.40 E38 to 3.40 E38
4 byte count value	0 to 4,294,967,295
6-byte fault information	to KNX specification, DPT_Alarminfo
8-byte system time	to KNX specification

Alarminfo:

AlarmPriority:	0x03 ≥ will not be analysed
ApplicationArea	0xFF = will not be analysed
ErrorClass	0xFF = will not be analysed

ApplicationArea and ErrorClass will be analysed for the event opening conditions only "via Value = Reference value".

8. COMMUNICATION OBJECTS (CONTINUED)

8.2 Event program communication objects (continued)

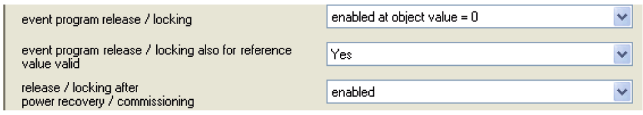
Event program 1

Parameter	Settings
event program release / locking	always enable enabled at object value = 0 enabled at object value = 1

This parameter determines whether the event program for running is always enabled or whether it can be enabled or disabled via the object "EP 1: release / locking".
With the setting "enabled at object value = 0" the EP is enabled on receiving a "0" and disabled on receiving a "1".
With the setting "enabled at object value = 1" the EP is enabled on receiving a "1" and disabled on receiving a "0".
With the selection "always enabled", the EP is permanently enabled and the object "EP 1: Release/Locking" and the parameters "Event program Release/Locking also valid for reference value" and "Release/ Locking after power return/start-up" are not available.

Note:
There is the possibility to enable or disable Release/Locking during operation via an external display-/control unit or via the BUS.

Event program 1



event program release / locking	enabled at object value = 0
event program release / locking also for reference value valid	Yes
release / locking after power recovery / commissioning	enabled

Parameter	Settings
Event program release / locking also for reference value valid	Yes No

This parameter determines whether the parameter "Event program Release/Locking" is also valid for the comparator value.
If the setting is "No", the reference value can always be updated, if the setting is "Yes", the reference value is ignored.

Event program 1

Parameter	Settings
release / locking after power recovery / commissioning	enabled disabled how before power recovery / release how before power recovery / locked

This parameter determines whether the event program for running after a power recovery and after start-up is enabled or disabled.
With the setting "how before power recovery/release", the status is set after a power recovery and its return to that extant before the power failure. The event program is enabled after start-up or after downloading. With the setting "how before power recovery/locked", the status is set after a power recovery and its return to that extant before the power failure. The event program is disabled after start-up or after downloading.

Event program 1

Parameter	Settings
event program retriggerable	No Yes

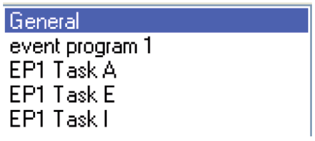
This parameter determines whether a running event program is stopped and restarted on receiving a fresh event opening and fulfilled condition during processing, or whether the fresh event opening is ignored and all event tasks are processed first.
This works if the event tasks in an EP are sent with different delay times.
If the openable retriggerable setting is "Yes", the running EP, in which not all event tasks are processed yet owing to a set sending delay, is stopped and restarted in the event of a fresh call-up.

Event program 1

Parameter	Settings
Number of event tasks	1 (A) 2 (A and B) 3 (A to C) 4 (A to D) 5 (A to E) 6 (A to F) 7 (A to G) 8 (A to H) 9 (A to I) 10 (A to K)

This parameter determines the number of event tasks which are used in event program 1. If this is selected, additional parameters and communication objects are created for the relevant event tasks.

Event program 1



Depending on the number of event tasks to be used, these task settings will be displayed in a maximum of 3 windows,
EP1: Task A…, EP1: Task E…, EP1: Task I…

EP1: Task A…

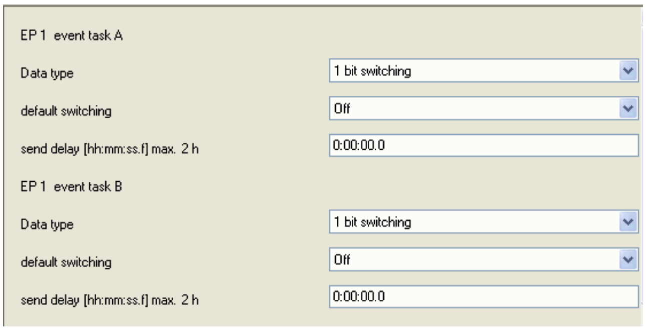
Select EP1: Task A… to select the settings up to task D.

EP1: Task E…

Select EP1: Task E… to select the settings up to task H.

EP1: Task I…

Select EP1: Task I… to select the settings up to task K.



EP 1 event task A

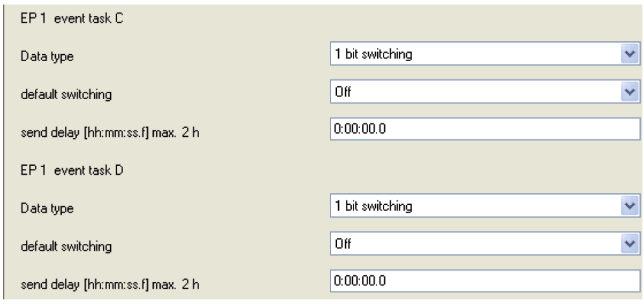
Data type	1 bit switching
default switching	Off
send delay [hh:mm:ss.f] max. 2 h	0:00:00.0

EP 1 event task B

Data type	1 bit switching
default switching	Off
send delay [hh:mm:ss.f] max. 2 h	0:00:00.0

8. COMMUNICATION OBJECTS (CONTINUED)

8.2 Event program communication objects (continued)



EP 1 event task C

Data type: 1 bit switching

default switching: Off

send delay [hh:mm:ss.f] max. 2 h: 0:00:00.0

EP 1 event task D

Data type: 1 bit switching

default switching: Off

send delay [hh:mm:ss.f] max. 2 h: 0:00:00.0

Parameter	Settings
Data type	1 bit switching 1 bit venetian blind 1 bit scene 2 bit forced control 4 bit dimming 1 byte value % 1 byte count value 8-bit scene recall 8-bit scene program 1 byte ASCII character 2 byte floating point 2 byte count value 4 byte floating point 4 byte count value 6 byte fault information 14 byte character string
This parameter determines the data type for the communication object "EP 1: Event task A". This provides the option to send a BUS telegram with any value. **Note:** The "14 bytes character string" is always present only in the relevant task A of event programs 1 to 8. The "Fault information" data type is always present only in the relevant task B of event programs 1 to 8.	

Different value ranges are allowed for the following parameters, depending on the task data type set.

EP1: Task A…

Parameter	Settings
default switching	Off On
default venetian blind	Up Down
default 1-bit scene	Scene A Scene B
default forced control	Forced control Off (0) Force controlled to Off (2) Force controlled to On (3)
default dimming relative	Stop (0) 100% darker (1) 1/2 darker (2) 1/4 darker (3) 1/8 darker (4) 1/16 darker (5) 1/32 darker (6) 1/64 darker (7) 100% brighter (9) 1/2 brighter (10) 1/4 brighter (11) 1/8 brighter (12) 1/16 brighter (13) 1/32 brighter (14) 1/64 brighter (15)
default value in %	0
default count value	0
default 8 bit scene	1 2 … 64
default ASCII characters	0
default floating point	0
default floating point	0
default 4-byte floating point	0
default 4-byte count value	0
default 6-byte fault information	0x00:0:00:00:00:00
default 14-byte character string	0

Use this parameter to set the initial default value of event task A in event program 1.

The value is maintained even after a BUS power failure and cannot be changed via the BUS.

Note:

Entries are limited to counts in the maximum possible value range in each case:

Value in %	0 to 100
1-byte count value	0 to 255
1-byte ASCII character	Entry (a character)
2-byte floating point	-671,088.64 to 671,088.96
2-byte count value	0 to 65,535
4-byte floating point	-3.40 E38 to 3.40 E38
4-byte count value	0 to 4,294,967,295
6-byte fault information	to KNX specification, DPT_Alarminfo
14-byte character string	Text input, max. 14 characters

8. COMMUNICATION OBJECTS (CONTINUED)

8.2 Event program communication objects (continued)

Note:
The "14 bytes character string" is always present only in the relevant task A of event programs 1 to 8.
The "Fault information" data type is always present only in the relevant task B of event programs 1 to 8.

EP1: Task A…

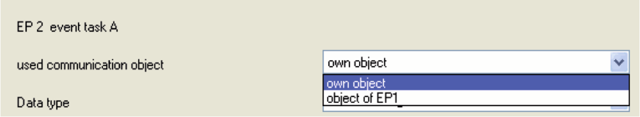
Parameter	Settings
send delay [hh:mm:ss.f] max. 2 h	0:00:00.0
This parameter determines now much time should elapse between the event opening and sending of event task A. This is how you can determine a given sequence when sending event tasks while running an event program. **Note:** The time entry is restricted to a value range between 0.1 seconds and 2 hours.	

The parameters for event tasks B to K are identical to those for event task A. They will only be displayed if they have been enabled via the "Number of event tasks" parameter.

Event programs 1…8

The parameters for event programs 1 and 8, for the relevant event tasks, are identical.
To reduce the setting overheads, there is an option, to be determined for each event task, whether proprietary sending objects or the relevant sending objects for the first event program in the allocated event range are to be used.
For this purpose, 8 event programs are divided into 2 event ranges. Each event range can be allocated a maximum of 4 event programs. You can allocate event programs 2, 3 and 4 to event program 1. Event program 1 thus forms the first event range. You can allocate event programs 6, 7 and 8 to event program 5.
Event program 5 thus forms the second event range.

Event tasks from event programs can only be allocated to objects from the first or fifth event program to which event tasks have also been set in the first or second event range, e.g. if event tasks A to E were created for EP 1, then only objects for event tasks A to E can also be allocated from event programs 2 to 4. Event task A from EP 2 uses the event task A object from EP 1.
Event task B from EP 2 uses the event task B object from EP 1, etc.
Accordingly, the event tasks for event programs 2, 3 and 4 have an additional parameter:



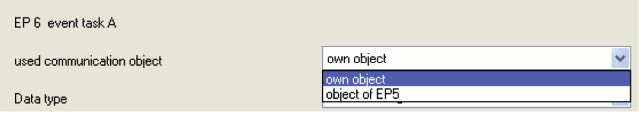
EP 2 event task A

used communication object: own object
own object
object of EP1

Data type

Parameter	Settings
used communication object	own object object from EP1
This parameter determines whether a proprietary object "EP 2: Event task A" is to be generated for event task A, or whether the object "EP 1: Event task A" is to be used. You can send several telegrams sequentially using a single communication object in this way.	

Accordingly, the event tasks for event programs 6, 7 and 8 have an additional parameter:



EP 6 event task A

used communication object: own object
own object
object of EP5

Data type

Parameter	Settings
used communication object	own object Object from EP5
This parameter determines whether a proprietary object "EP 6: Event task A" is to be generated for event task A, or whether the object "EP 5: Event task A" is to be used. You can send several telegrams sequentially in using a single communication object in this way.	