Software Function „SMIdrive-8“

KNX-Actuator for Control of SMI-Drives

Contents

General Description of Function

The SMIdrive-8 is an EIB-actuator for the control of up to 8 SMI-drives. Via the SMI (Standard Motor Interface) the drives will be addressed and controlled independently. Drive control is configured by parameters set up in the ETS (Engineering Tool Software from KNX) and accessed in operation via commands from the KNX-bus.

On a KNX-bus commands and data are transmitted via communication objects. The objects of the SMIdrive-8 provide independent control of the connected drives by standard functions as UP/DOWN, STEP/STOP and others. Furthermore status information as current blind- or Lamella-position in percent and drive errors can be requested via the bus. A hierarchy of prioritized operation states integrates manual and automatic actions as well as those controlled by safety functions or motion detention.

Commissioning of the device and setup of parameters is performed via the ETS. The ETS-interface provides the engineer with parameters organized in registers. The communication objects are displayed in table form. The addressing of the drives connected is part of the ETS-setup-procedure. On the SMI-bus-side drive addresses setup in the ETS-parameters are assigned either automatically or alternatively with respect to SMI-key-IDs input in parameter fields.

On the SMI-side the actuator restricts itself to use of commands from the common SMI-standard, thus ensuring compatibility with the SMI-drives of all manufacturers. For this purpose there is intentionally no functional support of manufacturer specific drive features. Nevertheless a manufacturer specific setup of drive features can be performed via the KNX-bus (see following paragraph).

Additional to the parameter setup via the ETS the SMIdrive-8 provides a further interface (so called user-interface-objects, short the UIO-interface) for commissioning, diagnosis and setup. This supports access to the internal configuration of the actuator and allows for direct access to the SMI-communication. To make use of these features a tool-software is provided, but generally a KNX-tool as the Device-Editor (which is part of the ETS3-package or available as APP for the ETS4) can as well be used to access this interface. The UIO-interface supports more detailed diagnoses during commissioning or modification of parameters in the SMI-drives themselves.

Software and Functionality

The Software of the SMIdrive-8 is made up of separate parts that share the common resources of the actuator (a microprocessor with integrated flash-ROM, RAM and additional EEPROM).

- The KNX-interface-system for the actuator called „System 2“ is a KNX-stack compatible to the BCU2-standard. It provides the complete interface to hardware resources and application program (firmware) on the one hand and to the KNX-bus on the other. A boot-loader in the Flash-ROM of the microprocessor is included in System 2 and supports updates of parts of the software, specifically the firmware (which is the application software including the SMI-communication library) by access via the KNX-bus. The firmware is mirrored in a copy in the EEPROM. The boot-loader will check the firmware-file, and only if it finds it valid, it will copy it from the EEPROM into the flash-ROM of the microprocessor and branch into its execution.
- An SMI-communication library is part of the firmware. It has been certified by the SMI-group and provides the interface to the SMI and thus the slaves (drives) connected to it.
- The application software itself uses the remaining memory (ROM) for the software-functionality of the actuator with regard to the ETS-parameter setup. It is this software part that also provides the support of the UIO-interface accessed via according system functions of the KNX- standard.
- The ETS-interface contains all the information on KNX-communication objects and parameters that govern the KNX-bus-interface and the functionality of the actuator. With the ETS these elements can be configured and loaded into the actuator. This way they are stored in the EEPROM of the actuator

and can then be accessed by the System 2 KNX-stack and by the firmware to be evaluated for determining the intended actuator behavior.

Software-tools provide access to specific memory regions oft he actuator via the KNX-bus. Besides standard development tools there is a specific firmware-update-tool, that under appropriate conditions allows replacement of the firmware by e.g. a different version, and there is an SMI-tool, bound for diagnostic purposes and direct access to the SMI via a gateway provided by the firmware via the UIO-interface.

In the successive paragraphs first the actuator-functionality as defined by the setup from the ETS-interface is described. Further chapters then explain the specifics for commissioning and setup and the UIO-interface.

Operating States

State Control and State Transition Table

The SMIdrive-8 principally distinguishes the following states (listed in the order of priority, highest first):

1) Motion detention
2) Safety
3) Manual operation
4) Automatic operation

Automatic operation can seperately be disabled for each drive by an ETS-parameter. In this case objects for switching from manual to automatic operation and vice versa do not appear in the list. Furthermore the objects for positioning blind-height and lamella-angle in % change their context to be read as manual commands for all channels (drives) with automatic mode disabled, while otherwise they have the rank of automatic-commands.

The state last present before a switching off of the KNX-supply-voltage will be reestablished after the KNX-supply-power is switched on again.

Motion Detention

This state has the highest priority. All drive movements are disabled in this state, even those required from a safety alarm-object. Setting the motion detention is bound e.g. for use to protect personnel from injury, who is busy in the area of potentially automatically moved mechanics (window cleaners for instance).

When activating the motion detention an eventually ongoing movement of the sun protection will be carried out to completion, but no further command-object requesting a movement will take effect if it is received after the detention-object.

As long as the motion detention is present, either one of the other states is active but “hidden” in the background. When the detention is redrawn, the hidden state will be reestablished and in the case of a safety-situation or in automatic operation the according movement will be carried out. The “hidden state” may be the one prevailing at the time the detention object was received or one that was calmly entered in the background by a telegram received in the meantime.

If for instance the prevailing state is automatic mode at the time the motion detention is being activated, then received automatic-mode-commands will be stored and the last resulting position will take effect after the detention has been redrawn. If during the state of detention a safety-object with the value “1” (meaning safety activation) is received, the hidden state switches to “safety”, i.e. the blind position that may be configured for this state will be moved to once the detention has been reset.

Manual mode commands received during an active motion detention will on the other hand just be ignored and thus will neither induce any “hidden” state switching.

Safety Mode

The safety-state has the second highest priority. When a safety-object with value “1” is received, the state of the actuator switches to “safety” (in case of a present motion detention the safety-state is “hidden” in background). If no motion-detention prevails, the actuator will move all drives into the respective safe position, if such has been configured. The safety-state is meant to be used for protection of the blind mechanics itself, e.g. against excessive wind or the like.

When the safety-state is entered, an eventually ongoing movement of blinds will be suspended and all further movement requests of automatic- or manual-state commands won’t take effect any more.

As long as the safety-state is active, one of the manual or automatic states hiddenly remains active in the background. If the hidden state is automatic, it will be reestablished when the sfety-state is reset and drives will be moved to the respective position. The hidden state is the one that prevailed, before the safety state was entered.

Commands for movements in automatic-mode received during safety-state will be stored to take effect, once the safety-state is reset, given that neither motion detention nor manual mode contradict with higher priority.

Manual-mode commands received while in safety-state will be ignored, and they do not either change the “hidden state”.

For channels, for which the automatic mode has been disabled by the according ETS-parameter, an additional parameter may be used to define, that after a safety-state-reset these shall (re-) establish positions from according manual-mode commands in % either as prevailing before the safety or as received and stored hiddenly during the safety-state. In the state-transition-table below an according note references the description here in connection with the state transition “safety-reset”. For the respective parameter refer to the parameter table below.

If the actuator receives an object Motion-detention with value “1” during a prevailing safety-state, the state is switched to detention and the safety-state turns to “hidden”.

Manual Mode Operation

The manual-mode state is assigned the third highest priority. Unless inhibited by a state of safety or motion detention, manual-mode commands will be executed immediately. If automatic mode prevails at the time the manual command is received, the state will additionally switch to manual-mode. A reset from manual mode to automatic mode on the other hand, can only be achieved by issuing an object “automatic” with value “1”.

Automatic Mode Operation

The automatic state has the lowest priority. Commands of automatic-mode will only be executed if the actuator is in automatic state.

State Transition Table

The following table lists the transitions between states with respect to events (as received telegrams or elapsed countdown time). The state “automatic mode” prevails as long as neither of the other states (manual mode, safety or motion detention) are active.

	Operating state			**Events**													
No	Motion-detention	Safety	Auto-matic	KNX-bus OFF	KNX-bus ON	Obj Motion-Detention ON	Obj Motion-Detention OFF	Obj. or timeout Safety ON	Obj. Safety OFF	Obj. Automatic ON	Obj. Automatic OFF	Obj Manually UP/DOWN	Obj Manually Step UP/DOWN	Obj. Recall Scene	Obj. Store Scene (drive at rest)	Obj. Blind-/ Lamella %-Position	Obj. UP/ Shading-Pos. (see note)
0	Off	Off	Off	**0** Move to Position for Bus-Failure	**0**	**4** Complete Ongoing Movement	**0**	**2** Move to Safety-Position	**0**	**1** Move to Stored Blind-/Lamella Position	**0**	**0** Move UP or DOWN respectively	**0** Step UP or DOWN respectively	**0** Establish Scene-Position	**0** Store Scene-Position	**0** Set Position only Stored	**0** Set Position only Stored
1	Off	Off	On	**1** Move to Position for Bus-Failure	**1** Move to Stored Blind-/Lamella Position	**5** Complete Ongoing Movement	**1**	**3** Move to Safety-Position	**1**	**1**	**0**	**0** Move UP or DOWN respectively	**0** Step UP or DOWN respectively	**0** Establish Scene-Position	**1** Store Scene-Position	**1** Set Position Stored & Established	**1** Set Position Stored & Established
2	Off	On	Off	**2** Move to Position for Bus-Failure	**2** Move to Safety-Position	**6** Complete Ongoing Movement	**2**	**2**	**0** see note in Operation States and Safety	**3**	**2**	**2**	**2**	**2**	**2** Store Scene-Position	**2** Set Position only Stored	**2** Set Position only Stored
3	Off	On	On	**3** Move to Position for Bus-Failure	**3** Move to Safety-Position	**7** Complete Ongoing Movement	**3**	**3**	**1** Move to Stored Blind-/Lamella Position	**3**	**2**	**2**	**2**	**2**	**3** Store Scene-Position	**3** Set Position only Stored	**3** Set Position only Stored
4	On	Off	Off	**4**	**4**	**4**	**0**	**6**	**4**	**5**	**4**	**4**	**4**	**4**	**4** Store Scene-Position	**4** Set Position only Stored	**4** Set Position only Stored
5	On	Off	On	**5**	**5**	**5**	**1** Move to Stored Blind-/Lamella Position	**7**	**5**	**5**	**4**	**4**	**4**	**4**	**5** Store Scene-Position	**5** Set Position only Stored	**5** Set Position only Stored
6	On	On	Off	**6**	**6**	**6**	**2** Move to Safety-Position	**6**	**4**	**7**	**6**	**6**	**6**	**6**	**6** Store Scene-Position	**6** Set Position only Stored	**6** Set Position only Stored
7	On	On	On	**7**	**7**	**7**	**3** Move to Safety-Position	**7**	**5**	**7**	**6**	**6**	**6**	**6**	**7** Store Scene-Position	**7** Set Position only Stored	**7** Set Position only Stored

ETS-Interface

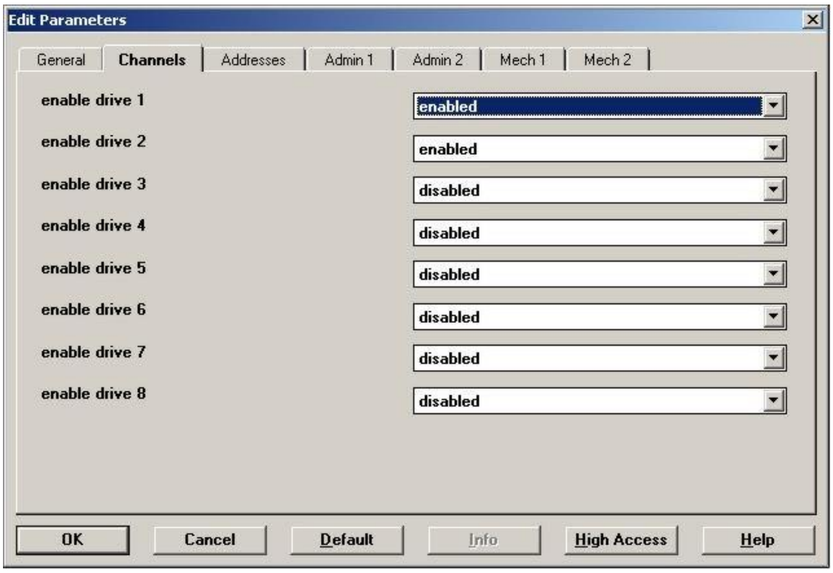
The layout of the ETS-interface (communication objects and parameters) has been designed similarly to typical sun protection actuators available on the market. This is to ease usage and commissioning of the device by integrators with minimum effort to familiarize themselves.

Parameters are ordered systematically in folders. Besides common parameters with effect on the actuator behavior as a whole, in two further folders the selection and addressing of channels present and respective drives connected can be defined. All further folders of parameters and according communication objects will be visible or not in accordance with respective parameter settings. As a default, all of the up to 8 channels are set to “disabled”. Thus none of the related communication objects will be present, but just the two common ones “Movement detention” and “Safety”, and no further parameter folders are visible. Additional folders and communication objects will pop up only as the respective channels are set to “enabled”.

If in the folder “General” the parameter “Drive configuration” is set to “all together”, two further folders will pop into life, called “Administration” and “Mechanics”. Settings in these two folders will affect the drive behavior for all channels in identical way. If “Drive configuration” is set to “individually”, one folder of its own with settings for administration and one with those for mechanics will be displayed for each channel set to “enabled”. Communication objects for channels enabled will accordingly pop up as the parameter settings require these.

Depending on the setting of “Addressing mode” in folder “General” there will only be one folder for definition of slave-addresses (with addressing mode “automatic”) or this slave-addresses-folder and two additional ones for the settings of manufacturers and key-IDs for the respective channels (with the addressing mode “by manufacturer and key-ID”).

In the following tables of communication objects and parameters, a note in column “depends on” will eventually inform, which parameter setting the respective object- or parameter-presence may depend of. The general dependency however, that objects and parameters will only be present for channels that have been enabled, will be taken as granted and not be explicitly mentioned any more.



enable drive 1 enabled

enable drive 2 enabled

enable drive 3 disabled

enable drive 4 disabled

enable drive 5 disabled

enable drive 6 disabled

enable drive 7 disabled

enable drive 8 disabled

figure 1:

ETS-interface, folders displayed

(ETS2, two channels enabled, individual configuration, automatic addressing)

Configuration Limits in ETS

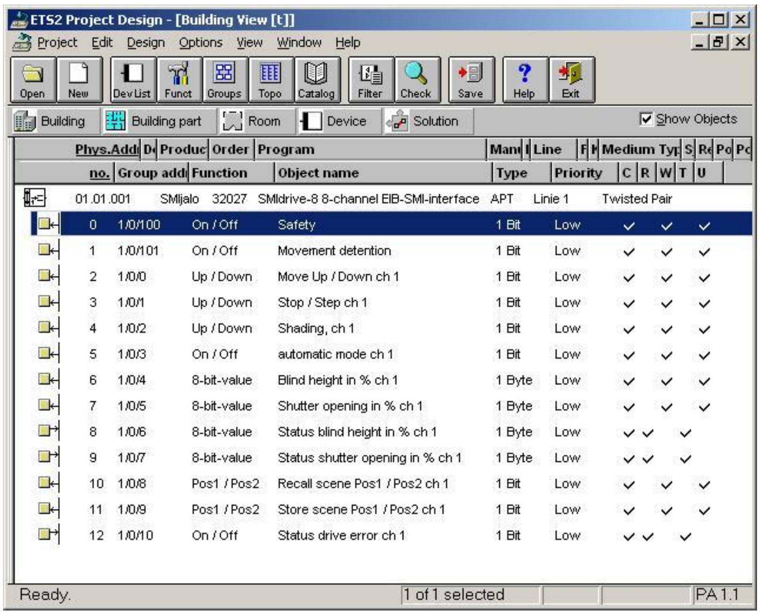
The actuator has 90 communication objects. With hese available for ETS-configuration are

114 group addresses and

122 associations.

ETS - Communication Objects

Obj.-No.	Object name	Operation mode	Depends on	Function	Type	Obj.-Flags
common objects (effective for all channels together)						
0	Safety			On / Off	1 Bit	CWU / rcv.
1	Movement detention			On / Off	1 Bit	CWU / rcv.
channel specific objects channel 1						
2	Move Up / Down ch 1	manual		Up / Down	1 Bit	CWU / rcv.
3	Stop / Step ch 1	manual		Up / Down	1 Bit	CWU / rcv.
4	Shading ch 1	automatic	Automatic mode enabled	Up / Down	1 Bit	CWU / rcv.
5	Automatic mode ch 1		Automatic mode enabled	On / Off	1 Bit	CWU / rcv.
6	Blind height in % ch 1	auto / man		8-bit value	8 Bit	CWU / rcv.
7	Shutter opening in % ch 1	auto / man	Venetian blind	8-bit value	8 Bit	CWU / rcv.
8	Status blind height in % ch 1			8-bit value	8 Bit	CRT / send
9	Status shutter opening in % ch 1		Venetian blind	8-bit value	8 Bit	CRT / send
10	Recall scene pos1 / pos2 ch 1	manual		pos1 / pos2	1 Bit	CWU / rcv.
11	store scene pos1 / pos2 ch 1	manual		pos1 / pos2	1 Bit	CWU / rcv.
12	Status drive error ch 1			On / Off	1 Bit	CRT / send
channel specific objects channel 2 … 8						
13	Move Up / Down ch 2	manual		Up / Down	1 Bit	CWU / rcv.
14	Stop / Step ch 2	manual		Up / Down	1 Bit	CWU / rcv.
15	Shading ch 2	automatic	Automatic mode enabled	Up / Down	1 Bit	CWU / rcv.
16	Automatic mode ch 2		Automatic mode enabled	On / Off	1 Bit	CWU / rcv.
17	Blind height in % ch 2	auto / man		8-bit value	8 Bit	CWU / rcv.
18	Shutter opening in % ch 2	auto / man	Venetian blind	8-bit value	8 Bit	CWU / rcv.
...	...	...	...	...	...	...
...	...	...	...	...	...	...
85	Status blind height in % ch 8			8-bit value	8 Bit	CRT / send
86	Status shutter opening in % ch 8		Venetian blind	8-bit value	8 Bit	CRT / send
87	Recall scene pos1 / pos2 ch 8	manual		pos1 / pos2	1 Bit	CWU / rcv.
88	store scene pos1 / pos2 ch 8	manual		pos1 / pos2	1 Bit	CWU / rcv.
89	Status drive error ch 8			On / Off	1 Bit	CRT / send


ETS2 Project Design - [Building View [t]]
Project Edit Design Options View Window Help
Open New Dev List Funct Groups Topo Catalog Filter Check Save Help Exit
Building Building part Room Device Solution
☑ Show Objects
Phys.Addı Dı Produc Order Program Manı I Line F I Medium Typ S Re Po Po

01.01.001 SMIjalo 32027 SMIdrive-8 8-channel EIB-SMI-interface APT Linie 1 Twisted Pair

no.	Group addı	Function	Object name	Type	Priority	C	R	W	T	U
0	1/0/100	On / Off	Safety	1 Bit	Low	✓		✓		✓
1	1/0/101	On / Off	Movement detention	1 Bit	Low	✓		✓		✓
2	1/0/0	Up / Down	Move Up / Down ch 1	1 Bit	Low	✓		✓		✓
3	1/0/1	Up / Down	Stop / Step ch 1	1 Bit	Low	✓		✓		✓
4	1/0/2	Up / Down	Shading, ch 1	1 Bit	Low	✓		✓		✓
5	1/0/3	On / Off	automatic mode ch 1	1 Bit	Low	✓		✓		✓
6	1/0/4	8-bit-value	Blind height in % ch 1	1 Byte	Low	✓		✓		✓
7	1/0/5	8-bit-value	Shutter opening in % ch 1	1 Byte	Low	✓		✓		✓
8	1/0/6	8-bit-value	Status blind height in % ch 1	1 Byte	Low	✓	✓		✓	
9	1/0/7	8-bit-value	Status shutter opening in % ch 1	1 Byte	Low	✓	✓		✓	
10	1/0/8	Pos1 / Pos2	Recall scene Pos1 / Pos2 ch 1	1 Bit	Low	✓		✓		✓
11	1/0/9	Pos1 / Pos2	Store scene Pos1 / Pos2 ch 1	1 Bit	Low	✓		✓		✓
12	1/0/10	On / Off	Status drive error ch 1	1 Bit	Low	✓	✓		✓	

Ready. 1 of 1 selected PA 1.1

figure 2:

communication objects in ETS

(one channel enabled, venetian blind, automatic mode enabled)

Object „Safety“

This object can be used to switch the safety function. It can be connected to a wind velocity guard module. For each channel a parameter defines if the safety function is enabled for this channel and what drive reaction is required in case of a safety alarm (Up-movement, Down-movement or no movement). The object itself affects all channels that have their safety function enabled. If the safety function is active and enabled from the respective parameter, all movement requests received for the respective channel are disabled. Movement requests will only be served again after the safety alarm has been reset.

object value „1“: Safety function ON

object value „0“: Safety function OFF

Object „Movement Detention “

A movement detention can be activated or deactivated by this object. For each channel a parameter defines, if the motion detention function is enabled or disabled fort e respective channel. The object itself affects all channels that have the motion detention function enabled. If the movement detention is activated and enabled for the respective channel, all movement requests received for this channel are disabled. Nevertheless a movement going on at the time the movement detention is activated will be carried out to its end.

object value „1“: Movement Detention ON

object value „0“: Movement Detention OFF

Object „Move Up / Down channel x” (manual mode)

This object will move the drive in the respective direction.

If the blind type parameter is set to “Venetian Blind” and the parameter „open shutter after manual down“ is set to „enabled“, then a movement to the lower most position will be followed by an opening of the shutter lamellas by the value specified in the respective parameter. If the blind already is in the lower most position when the movement request is received only the shutter lamellas will be closed and not be opened again.

object value „0“: Upward movement

object value „1“: Downward movement

Object „Stop / Step channel x“ (manual mode)

With this object either an ongoing movement will be stopped or the drive will be stepped by a defined angle.

object value „0“: Stop or step in up direction

object value „1“: Stop or step in down direction

Object „Shading channel x“ (automatic mode)

With this object the drive can be moved Up or Down similar to the object „move“. Additionally the drive will open the shutter lamellas by the defined angle after having reached the lower most position.

This object is only present, if the automatic mode has been enabled for the channel.

object value „0“: Upward movement

object value „1“: Downward movement

Object „Automatic mode, channel x“

With this object the operation mode of the actuator can be switched from automatic to manual and vice versa. A switching of the operation mode can by priority-order of operation modes also be effected by different objects. See State Control and State Transition Table in this context.

This object is only visible, if automatic operation has been enabled for the respective channel.

Object value „0“: Automatic mode OFF

Object value „1“: Automatic mode ON

Object „Blind height in %, channel x“ (automatic mode or, if automatic mode disabled, manual mode)

With this object and a 1-byte-value (0-255) the blind height can be positioned scaled with respect to its maximum range. A lamella angle that has been set automatically in % before will be restored afterwards.

If automatic operation has been enabled for the respective channel, the object „blind height“ will only take effect if the automatic mode is active, otherwise the object value will be stored in the actuator and the position will be established as soon as the automatic mode has been activated by the respective object. If the automatic mode has been disabled for the respective channel, this object will be interpreted as a manual mode object and take effect immediately.

Object value „0“ (0%): Move blind height to upper most position

Object value „255“ (100%): Move blind height to lower most position

All values in the range from 0 through 255 are valid.

Object „Shutter opening in %, channel x“ (automatic mode or, if automatic mode disabled, manual mode) (only for venetian blinds)

With this object and a 1-byte-value (0-255) the angle of lamellas (shutter opening) can be positioned scaled with respect to its maximum range. The blind height will remain unchanged.

If automatic operation has been enabled for the respective channel, this object will only take effect if the automatic mode is active, otherwise the object value will be stored in the actuator and the lamella angle will be established as soon as the automatic mode has been activated by the respective object. If the automatic mode has been disabled for the respective channel, this object will be interpreted as a manual mode object and take effect immediately.

This object is only visible, if the blind type has been set to venetian blind for the respective channel.

Object value „0“ (0%): Lamella horizontal / open

Object value „255“ (100%): Lamella closed

All values in the range from 0 through 255 are valid.

Object „Status blind height in %, channel x“

With this object the current blind height can be requested and read in %. The value will be responded as 1 byte scaled to a range from 0 (0%) through 255 (100%). The actuator will send this status without request, if the general parameter „send status objects“ has been set to „upon state change“.

Object value „0“ (0%): upper most position

Object value „255“ (100%): lower most position

All values in the range from 0 through 255 may be given as response.

Object „Status shutter opening in %, channel x“ (only for venetian blinds)

With this object the current state of the shutter opening (lamella angle) can be requested and read in %. It will be responded as a 1-byte-value scaled to the range from 0 (0%) through 255 (100%). The actuator will send the status without request, if the general parameter „send status objects“ has been set to „upon state change“.

This object is only visible, if the blind type has been set to venetian blind for the respective channel.

Object value „0“ (0%): Lamella horizontal / open

Object value „255“ (100%): Lamella closed

All values in the range from 0 through 255 may be given as response.

Object „Recall scene pos1 / pos2, channel x“ (manual mode)

With this object the positions pos1 or pos1 can be recalled, which have been stored in the actuator before. For venetian blinds both blind height and shutter opening (lamella angle) will be restored, for simple blinds only the blind height will be restored. For venetian blinds depending on the direction in which the respective height has to be established, the actuator will by consecutive movements make sure that the resulting positions of height and shutter will both resemble the ones which prevailed when the position was stored.

Object value „0“: restore pos1

Object value „1“: restore pos2

Object „Store scene pos1 / pos2, channel x“ (manual mode)

With this object the currently prevailing position of blind height and shutter opening (when controlling venetian blinds) can be stored in the actuator as pos1 or ops2. Anyway the actuator will only store the position if the drive is at rest at the time the command telegram is received.

Object value „0“: store position as pos1

Object value „1“: store position as pos2

Object „Status drive error, channel x“

With this object an eventual drive error state can be requested and read. The actuator will send this status without request, if the general parameter „send status objects“ has been set to „upon state change“.

Object value „0“: no drive error

Object value „1“: drive has signaled an error

Note: Whenever it makes sense the firmware of the actuator will attempt to assemble motion commands (e.g. „move up / down, channel x“) that have been received in a single telegram for more than one channel (if for instance received with the same group address) into according SMI-group-commands for the respective channels. In this way a synchronous operation of the connected drives can be made sure. Upon reception of Step-commands the drives will additionally be grouped by according same value of the configured step length parameter.

ETS-Parameters

(Default values of parameters are set in **bold**)

Parameter	Setting	Description	Dependency
Folder „general“			
Send status objects	**Only upon request** upon status change	This parameter can be used to configure if the status-objects “blind height“, “shutter opening“ and “drive error“ of all channels will only be sent upon a reading request or if they will be sent without a request as soon as a new position has been established or when a drive has signaled or reset an error status.	
upon bus recovery	Do not send anything **Send status objects**	This parameter can be used to configure if after a recovery of the bus voltage the Status objects “blind height“, “shutter opening“ and “drive error“ of all channels will be sent. The transmission will eventually be delayed by a time that depends on the physical address of the actuator, a feature that is meant to help to avoid bus conflicts by too many bus components transmitting at the same time.	Send status objects = “upon status change“
Safety	**without timeout** with timeout	This parameter can be used to configure if the reception of safety telegrams with value “safe” in regular intervals shall be checked with a timeout.	
Timeout limit for safety	1 minute 5 minutes **10 minutes** 30 minutes	If the parameter “safety“ has been set to “with timeout control“ this parameter can be used to configure the timeout, a reception of the safety object with the value “safe” is expected regularly.	Safety = “with timeout“
Drive configuration	**individually** all together	This parameter can be used to configure if channels (drives) shall be configured individually or with same settings all together. If set to “individual“ the respective parameter folders will pop up for each channel that has been activated. If set to “same for all“ the parameter folders are restricted to a single set and parameters therein then govern the behavior of all drives the same.	
Adressing mode	**automatic** by manufacturer and key-ID	If this parameter is set to “automatic” the actuator will automatically assign the slave-addresses given by ETS-parameters to all SMI-slaves (drives) that can be programmed (i.e. those with initial slave-Address 0). The largest SMI-slave-address will then be assigned tot he drive with the largest SMI-key-ID and so forth. If set to “by manufacturer and SMI-key-ID” the actuator will assign slave addresses only according to settings in respective ETS-parameters.	
readdressing	**automatic** force	This parameter can be used to initiate a readdressing of all connected SMI-slaves (drives).	
Parameter Folder “Channels“			
enable drive 1	**enabled** disabled	This parameter can be used to configure if a drive shall be controlled by this channel.	
enable drive 2	**enabled** disabled	This parameter can be used to configure if a drive shall be controlled by this channel.	
…	…	…	…

enable drive 8	**enabled** disabled	This parameter can be used to configure if a drive shall be controlled by this channel.	
Parameter Folder “Addresses“			
SMI-slave-address of drive 1	0, **1,** 2, ..., 15	This parameter can be used to configure the SMI-slave-address to be used to access the drive connected.	enable drive 1 = “enabled“
SMI-slave-address of drive 2	0, 1, **2,** ..., 15	This parameter can be used to configure the SMI-slave-address to be used to access the drive connected.	enable drive 2 = “enabled“
…	…	…	…
SMI-slave-address of drive 8	0, 1, 2, ..., **8,** ..., 15	This parameter can be used to configure the SMI-slave-address to be used to access the drive connected.	enable drive 8 = “enabled“
Parameter Folder “Manufacturer“			Adressing mode = “by manufacturer and SMI-key-ID“
Manufacturer code drive 1	**any,** Alcatel, Becker, Elero, Selve, 5, Vestamatic, Varema, 8, 9, ..., 15	If the addressing mode is set to "by manufacturer and key-ID“, this parameter is used to select the manufacturer of the drive to be controlled by this channel.	and enable drive 1 = “enabled“
Manufacturer code drive 2	**any,** Alcatel, Becker, Elero, Selve, 5, Vestamatic, Varema, 8, 9, ..., 15	If the addressing mode is set to "by manufacturer and key-ID“, this parameter is used to select the manufacturer of the drive to be controlled by this channel.	and enable drive 2 = “enabled“
…	…	…	…
Manufacturer code drive 8	**any,** Alcatel, Becker, Elero, Selve, 5, Vestamatic, Varema, 8, 9, ..., 15	If the addressing mode is set to "by manufacturer and key-ID“, this parameter is used to select the manufacturer of the drive to be controlled by this channel.	and enable drive 8 = “enabled“
Parameter Folder „SMI-key-IDs“			Adressing mode = “by manufacturer and key-ID“
SMI-key-ID drive 1	SMI-key-ID of the drive (**0** ... 4294967295)	If the addressing mode is set to "by manufacturer and key-ID“, this parameter is used to set the SMI-key-ID of the drive to be controlled by this channel.	and enable drive 1 = “enabled“
SMI-key-ID drive 2	SMI-key-ID of the drive (**0** ... 4294967295)	If the addressing mode is set to "by manufacturer and key-ID“, this parameter is used to set the SMI-key-ID of the drive to be controlled by this channel.	and enable drive 2 = “enabled“
…	…	…	…
SMI-key-ID drive 8	SMI-key-ID of the drive (**0** ... 4294967295)	If the addressing mode is set to "by manufacturer and key-ID“, this parameter is used to set the SMI-key-ID of the drive to be controlled by this channel.	and enable drive 8 = “enabled“
Parameter Folder “Administration“			Drive configuration = “all together“
Safety (e.g. wind alarm)	enabled **disabled**	This parameter can be used to configure if the safety object and the safety function are effective for all channels.	

Movement upon safety	no action **move up** move down	This parameter can be used to configure if drives on all channels are to move in a defined direction or stay at present position upon safety alarm. Irrespective of this setting all motion commands received via KNX-bus are inhibited if the sefety function has been enabled and a safety alarm prevails.	and Safety = “enabled“
restore positions after safety reset	**disabled** enabled	If automatic mode has been disabled this parameter can be used to configure for all channels if %-positions received as manual mode telegrams shall be established or restored after a safety alarm is reset.	and automatic mode = “disabled“
Movement detention	enabled **disabled**	This parameter can be used to configure if the movement detention is effective for all channels. If movement detention is enabled and active, all motion commands received via the KNX-bus are inhibited.	
Action upon bus failure	no action **move up** move down	This parameter can be used to configure if upon a bus failure all drives are to move in a defined direction or to stay in present position.	
Action upon bus recovery	no action **move up** move down	This parameter can be used to configure if upon a bus recovery all drives are to move in a defined direction or to stay in present position.	
automatic mode	disabled **enabled**	This parameter can be used to configure if all channels support the automatic mode. If set to “enabled“ the objects “shading“, ”blind height“ and “shutter opening“ will take effect only if the automatic mode is active. Otherwise their values will be stored and take effect only upon later activation of the automatic mode. If set to ”disabled” the objects ”blind height“ and “shutter opening“ will be interpreted as manual mode objects.	
Parameter Folder „Mechanics“			Drive configuration = “all together“
Blind type	**Venetian blind** Blind	This parameter can be used to configure for all channels which kind of blind will be controlled.	
Angle factor for step (base: 2°)	0-255 **5**	This parameter can be used to configure the width of a step in units of 2° for all channels.	
Angle factor open shutter from shading (base: 2°)	0-255 **30**	This parameter can be used to configure by which angle the lamellas will be opened in the shading position for all channels.	and Blind type = “Venetian blind“
Angle factor open shutter from blind down (base: 2°)	0-255 **30**	This parameter can be used to configure by which angle the lamellas will be opened in the lower most position after a down-movement for all channels. For textile blinds this parameter can be used to tighten the cloth.	and Blind type = “Blind“
open from manual down	**disabled** enabled	This parameter can be used to configure if the shutter (lamellas) shall be opened by the angle set in parameter “Angle factor open shutter from blind down” after a manual down movement for all channels. In this case a shutter-opening will also be effected if a manual down-movement is stopped underway.	
Angle factor shutter open (base: 2°)	0-255 **45**	This parameter can be used to configure the angle from closed to the maximum shutter (lamella) opening for all channels.	and Blind type = “Venetian blind“

Angle factor from open to blind movement (base: 2°)	0-255 **30**	This parameter can be used for all channels to configure the angle the lamellas will move upon a move-up or successive step-up commands from the open position until the blind height starts to change.	and Blind type = “Venetian blind“
Parameter Folder „Admin #“ (# is the number of the channel 1 ... 8)			Drive configuration = “individually“ and enable drive # = “enabled“
Safety (e.g. wind alarm)	enabled **disabled**	This parameter can be used to configure if the safety object and the safety function are effective for the respective channel.	
Movement upon safety	no action **move up** move down	This parameter can be used to configure if the drive on the respective channel is to move in a defined direction or stay at present position upon safety alarm. Irrespective of this setting all motion commands received via KNX-bus are inhibited if the sefety function has been enabled and a safety alarm prevails.	and Safety = “enabled“
restore positions after safety reset	**disabled** enabled	If automatic mode has been disabled this parameter can be used to configure for the respective channel if %-positions received as manual mode telegrams shall be established or restored after a safety alarm is reset.	and automatic mode = “disabled“
Movement detention	enabled **disabled**	This parameter can be used to configure if the movement detention is effective for the respective channel. If movement detention is enabled and active, all motion commands received via the KNX-bus are inhibited.	
Action upon bus failure	no action **move up** move down	This parameter can be used to configure if upon a bus failure the respective drive is to move in a defined direction or to stay in present position.	
Action upon bus recovery	no action **move up** move down	This parameter can be used to configure if upon a bus recovery the respective drive is to move in a defined direction or to stay in present position.	
automatic mode	disabled **enabled**	This parameter can be used to configure if the respective channel supports the automatic mode. If set to “enabled“ the objects “shading“, ”blind height“ and “shutter opening“ will take effect only if the automatic mode is active. Otherwise their values will be stored and take effect only upon later activation of the automatic mode. If set to ”disabled” the objects ”blind height“ and “shutter opening“ will be interpreted as manual mode objects.	
Parameter Folder „Mech #“ (# is the number of the channel 1 ... 8)			Drive configuration = “individually“ and enable drive # = “enabled“
Blind type	**Venetian blind** Blind	This parameter can be used to configure which kind of blind will be controlled by the respective channel.	
Angle factor for step (base: 2°)	0-255 **5**	This parameter can be used to configure the width of a step in units of 2° for the respective channel.	
Angle factor open shutter from shading (base: 2°)	0-255 **30**	This parameter can be used to configure by which angle the lamellas will be opened in the shading position for the respective channel.	and Blind type = “Venetian blind“

Angle factor open shutter from blind down (base: 2°)	0-255 **30**	This parameter can be used to configure by which angle the lamellas will be opened in the lower most position after a down-movement for the respective channel. For textile blinds this parameter can be used to tighten the cloth.	and Blind type = “Blind“
open from manual down	**disabled** enabled	This parameter can be used to configure if the shutter (lamellas) shall be opened by the angle set in parameter “Angle factor open shutter from blind down” after a manual down movement for the respective channel. In this case a shutter-opening will also be effected if a manual down-movement is stopped underway.	
Angle factor shutter open (base: 2°)	0-255 **45**	This parameter can be used to configure the angle from closed to the maximum shutter (lamella) opening for the respective channel.	and Blind type = “Venetian blind“
Angle factor from open to blind movement (base: 2°)	0-255 **30**	This parameter can be used for the respective channel to configure the angle the lamellas will move upon a move-up or successive step-up commands from the open position until the blind height starts to change.	and Blind type = “Venetian blind“

Projecting and Commissioning

Initialization with Drives that already have Slave-Addresses programmed

During the initialization (after a bus-reset, powering-on or download from ETS) the actuator will attempt to access the slaves (drives) that should be present on each channel as defined in the ETS-parameter setup using the respective SMI-slave-addresses. If the actuator had been commissioned before it has manufacturer and key-ID data for all drives present stored in its nonvolatile memory.

For every slave (drive) that has been configured via ETS and which could be accessed on the SMI the manufacturer code and key-ID will be read and cross-checked against the entries stored. If the addressing mode has been set to “by manufacturer and key-ID” the respective data from ETS-setup will be included in the cross-checks. If the actuator does not detect any change in the data of slaves with respect to those present on the SMI during last initialization, it will be ready for operation then at once.

If automatic addressing mode is used and if drives have been replaced since the last initialization, but slaves on the SMI do respond when accessed with the respective slave-addresses, then the manufacturer-codes and key-IDs that eventually have changed will be stored accordingly updated inside the actuator. If the addressing mode is “by manufacturer and key-ID” then manufacturer and key-ID data read from the slaves must match those configured in the ETS parameters, otherwise the drives will not be accepted by the actuator and ignored in operation.

Drives on channels that have been configured as venetian blinds will be calibrated, if the actuator has detected them as new or replaced. For calibration the actuator will have the drive perform a movement of a fixed length of 500°, in order to use the difference of relative positions read from the drive before and after the movement to determine the relation between the relative SMI-positions and degrees. This ratio will later be used for the positioning of blind height and shutter opening (lamella angles) to make sure that any situation can be restored directly and accurately irrespective from which direction, while angles involved in lamella positioning have been configured in degrees (°) only.

In a first commissioning of the actuator with drives connected that come with slave-addresses readily programmed, there will be no new slave-address-programming by the actuator but only a calibration-movement as far as required.

Initialization with Drives that do not have Slave-Addresses Programmed (i.e. having Slave-Address 0)

If during initialization there is no response from a drive on one or more of the slave-addresses configured via ETS, the actuator will attempt to assign these “missing” slave-addresses to drives by programming. Two different addressing modes are supported:

With automatic addressing an algorithm from the SMI- standard is employed for a key-ID-search. When a programmable drive has been found (i.e. one with the slave-address 0) the actuator will assign it the respective slave-address from ETS-parameters and program the drive accordingly. The assignment will use the highest “free” slave address for the drive with the largest key-ID, then the next smaller one and so forth until no further address-assignment can be programmed.

If key-ID-addressing has been selected, the actuator will attempt to program every drive directly with the slave-address that has been assigned to the respective manufacturer and key-ID in the ETS setup. If this fails the channel with the respective slave-address will stay without function.

Errors and Warnings during Initialization

No matter which addressing mode is employed, the actuator will generate according error-codes if the assignment of a slave (drive) to a channel fails for some reason. Warnings will be generated if a slave is detected as replaced or if one was programmed newly with a slave-address. Always the last six different errors or warnings are stored in the respective property in the device-object of the actuator. Property 201 (hex C9) contains the warnings, property 202 (hex CA) the errors. Possible codes of errors and warnings and their respective reasons are described in the respective table in the section *Errors and Warnings in Properties 201 & 202 in the Device-Object* below.

Strategy when Projecting and Commissioning SMIdrive-8

Generally the best suitable of the addressing modes “automatic” or “by manufacturer and key-ID” should be selected depending on the availability of drive data (manufacturer and SMI-key-IDs), which may or may not have been recorded by the electrician during installation.

Many of the drive-manufacturers provide labels or removable stickers on drives, on cables or on the blind assemblies with a copy of the key-ID of the respective drive in order to support an addressing procedure using the key-IDs.

If the drive-data are clearly assigned to the drives in the different mounting locations, using an addressing mode “by manufacturer and key-ID” the commissioning engineer can accomplish commissioning with a single ETS-download per actuator. An additional advantage of this procedure is that in the end the relevant drive data will be recorded in the ETS-project.

If the drives have been programmed with different but known slave-addresses previously, the same applies even if the key-IDs are not available. Assignment of actuator channels to the drives in different mounting locations is then accomplished solely by assigning the slave-addresses properly in the ETS.

In all other cases the “automatic” addressing mode is preferable. Nevertheless, using this, the assignment of channels to the drives and blinds in different mounting locations is initially arbitrary. After figuring out the “who-is-who” then by test and rearranging the slave-addresses in the respective ETS-parameter-folder the final assignment is established with a second ETS-download.

If the drives have preprogrammed non-zero but unknown slave-addresses, or some may have while others have slave-address 0, then the ETS-parameter “reprogramming” in folder general may be set to “force”. This will have the actuator first delete the slave-addresses of all connected drives (i.e. reset them to 0), before it begins to acquire the key-IDs, assign the slave-addresses and program the drives accordingly following the selected addressing-mode. Depending on the latter the channel-assignment may be finished or require a second download after rearranging the slave-addresses as described above.

Options if Errors are Faced during Projecting and Commissioning

If during a complete new addressing or readdressing of drives the actuator determines more key-IDs (slaves) present on the SMI than are expected according to channels and slave-addresses enabled in the ETS-setup, the surplus drives are left with the slave-address 0. If on the other hand no further slave (drive) can be found on the SMI while there are one or more channels left enabled from the ETS-setup, which hence cannot be assigned, then these remaining channel or channels will stay without function.

Other possible errors during commissioning may eventually be identified by inspecting the codes in properties 201 and 202 from the device-object and decoding these referring to the respective table below (for instance the device-editor from the ETS3 package can be used to inspect properties in the KNX-objects).

In the same way, the properties in the user-interface-object “50001” can be inspected and interpreted referring to the description following below (section *User-Interface-Objects and Properties (UIO-Interface)*).

A diagnostic support-tool with convenient options – the SMI-tool – is provided free of charge as a download from the site www.rk-aht.de. With this tool the access to different actuators and the respective properties of the user-interface-object “50001” is facilitated and simplified considerably (see following sections).

User-Interface-Objects and Properties (UIO-Interface)

In the user-interface-objects and the properties defined therein an interface is provided that – by use of according tools – facilitates advanced diagnostics, settings and interventions via KNX-bus and specifically allows some means of access to the SMI of the respective actuator.

Use of the UIO-interface can be made by either a standard-tool as for instance the device editor (an accessory from the ETS3-package), or by a specific tool-software that can be used independently from the ETS, but which makes according use of the Falcon-library to access the KNX-bus and establish a communication with the respective actuators of type SMIdrive-8. The free-of-charge SMI-tool requires an installation of the Falcon-library to be present (as the Falcon-library is part of the ETS3-installation, it will be if an ETS3 is installed on the computer).

Use of the UIO-interface with a tool like the device editor nevertheless requires specific knowledge and some advanced experience of the user. This chapter should hence have been read carefully and understood comprehensively. Application of the device editor for diagnostic purposes is suggested only for urgent cases of otherwise inexplicable malfunctions. Much easier use of the UIO-interface can be made with the SMI-tool (see section *The SMI-Tool-Software*).

Additionally to the EIB/KNX-objects defined in the KNX-standard the actuator supports one further object with the ID “50001”, which provides functionality for diagnosis and commissioning of the SMI and the SMI-slaves connected to the actuator. The tool-software mentioned above visualizes the properties provided via the user-interface-object (UIO) 50001 for a clear diagnosis. It facilitates intervention into SMI-addressing of slaves (drives) and their configuration in a convenient way.

Generally the UIO-interface provides a way to access SMI-slaves directly from the KNX-bus and to issue any possible SMI-command addressing one or more of the drives on the SMI, the actuator serving as a gateway in this case. The gateway functionality is provided by the property 64. There is no dedicated hardware connection to the SMI required, since all actuators present are accessed from the KNX-bus and provide the connection to the respective SMI themselves. Communication from the KNX-bus is directed to the specific actuator (and the respective SMI) using a connection opened with the physical address of the actuator. In principle of course drive-internal parameters may be accessed and set in this way, too.

Errors and Warnings in Properties 201 & 202 in the Device-Object

The KNX-stack “System 2” implemented in the actuator also provides supplemental properties in the “device-object”, which is one of the four objects defined in the KNX-standard of BCU2-class mask-versions of bus-couplers. These properties, with the IDs 201 and 202, are controlled by the firmware of the actuator and are bound to facilitate some first means of access to errors or warnings the actuator may have generated if drive communication encountered some exception, anomaly or malfunction.

Both, the blind-actuator-firmware and the “System 2” KNX-stack will eventually write message codes into the properties 201 or 202 of the device-object. Messages from the stack can be identified by their codes, which have values ≤ hex 6F and are explained in the “system 2”-documentation. Message codes written by the blind-actuator have values ≥ hex 70 and can be found in the following table.

Code	Art	Bezeichnung	Ursache
0x73	error	SMI transmission-timeout	error in the SMI-communication during transmission attempt, eventually SMI-power is down
0x74	error	EEPROM-error	data-error in the EEPROM (checksum invalid), ETS-download potentially erroneous
0x76	error	error in slave-programming	the attempt to program a slave failed
0x77	error	duplicate slave-address	in the ETS a slave-address has been assigned twice (same for different channels)
0x78	warning	unexpected slave	actuator detected a slave, that was not expected from the ETS-setup
0x7a	warning	EEPROM was updated	one or more new or replaced slaves were identified

0x7b	warning	forced reprogramming	a slave reprogramming has been requested via the respective parameter in ETS-setup
0x7c	warning	missing slaves	there were more slaves configured in ETS-setup than could actually be identified and assigned on the SMI
0x7d	warning	slave ignored due to key-ID	a slave will be ignored, that has a key-ID different from those expected from ETS-setup

User-Interface-Object 50001 and Properties therein

Properties defined in the user-interface-object 50001 are listed in the table below and explained in detail in the text following.

Prop-ID	data type	length	R/W	memory	meaning / function
1	obj-ID	2 bytes	RO	Code	object-ID of the user-interface-object, fixed to 50001
51	version	2 bytes	RO	RAM	internal firmware-version
64	char[17]	17 bytes	R/W	RAM	gateway for SMI-commands
65	char[8]	8 bytes	RO	RAM	status of channels 1 … 8 (1 byte per channel)
80	int	2 bytes	RO	EEPROM	bit-mask of slaves on the SMI
81	char[6]	8 bytes	RO	EEPROM	channel-no (1 byte) drive-type & manufacturer-ID (1 byte) key-ID (4 bytes) calibration value (2 bytes) of slave 1
82	char[6]	8 bytes	RO	EEPROM	dto. of slave 2
...	...	...	...	...	...
87	char[6]	8 bytes	RO	EEPROM	dto. of slave 7
88	char[6]	8 bytes	RO	EEPROM	dto. of slave 8

Property 1

As defined in the KNX-standard property 1 reflects just the object-ID. It is read-only.

Property 51

Property 51 makes use of the 2-bytes-datatype PDT_VERSION from KNX-standard and can be used to read the internal firmware-version. Always upon initialization the firmware generates a read-only RAM-copy of the firmware-version designator that is coded fixed in the flash-memory of the controller. All four nibbles of the 2-byte designator have a meaning of their own: the first (ms-) byte designates major and middle version numbers in BCD-Coding, the second (ls-) byte carries a second-middle version and a minor version number. The second-middle version number is translated into letters starting with ‘A’ for value ‘1’ in the version designation, communicated to customers. Thus the hex coding 2232h would read as version “2.2 C 2”.

Property 64

Property 64 provides the gateway to the SMI-Bus – it is bidirectional for this reason, i.e. it can be read and written from outside. Its length of 17 bytes allows for a transmission of any valid SMI-command, which will be forwarded unmodified to the SMI by the actuator. From the SMI-protocol specification commands may actually carry up to 16 bytes. A checksum need not be appended into the property when filling in a telegram, this will be supplemented by the actuator firmware before forwarding it to the SMI. Anyway, one byte more is reserved for some overhead function: the first or left-most byte (per KNX-definition the one with index 0) serves for controlling the data transfer via the gateway-property. This control-byte can also be used to provoke special reactions in the actuator. It can be written with the values listed in the following table:

Value	Function
00	reset of the firmware, i.e. re-initialization of the actuator
01	SMI command in the following 16 bytes for forwarding to the SMI-bus
02	(re-) calibration of all slaves configured as venetian blind
03	(re-) calibration of a single slave, slave-address in the next byte (this command may be unavailable depending on firmware variant)

04	reset of the application, i.e. re-initialization of the actuator, but in contrast to option 00h above without updating position data in the EEPROM
08	Output of the values of all properties in user-interface-object 50001 to the diagnostic interface

If from outside a value 01h is written into the control byte, the actuator will interpret the following 16 bytes as an SMI-telegram to be forwarded, will generate and append the proper check-sum and transmit the telegram via SMI, if necessary including repetitions as required according to SMI-standard.

All values from the above table that may be written into the control-byte (except 00h and 04h for re-initializations) will be acknowledged immediately with proper response by the actuator in the same place, i.e. in the control byte. Polling the control byte can thus be used for monitoring the execution state of the respective request. Principally the value FFh will in this context be interpreted as ACK (analogously to the SMI-protocol-standard).

When the actuator performs a re-initialization, the assignments of slaves (drives) to channels may change. The properties 65 and 80 though 88 will then be updated accordingly. The actuator will leave the value in the control byte untouched for the time the whole process takes, and will set it to FFh once the initialization is finished. Thus in this case the value FFh also indicates validity of above mentioned properties after according update during re-initialization to the user of the UIO.

A recalibration-request will be responded to by the actuator at first with a value of 12h in the control-byte. Only after completion of the recalibration will the actuator acknowledge this status with the value FFh. Depending on number, type and configuration of the drives concerned, the recalibration may take some time for completion.

The actuator-response to a request to forward an SMI-telegram will in the first place be a value of 08h in the control-byte, which indicates that the telegram is being transmitted on the SMI. As soon as a response to this telegram (which of course may be a slave-response) has been received on the SMI, this response will be reflected in the bytes of the property 64 starting from byte-index 1, while the control-byte will take on one of the values from the following table:

Wert	Bedeutung
FFh	slave has responded with ACK
E0h	slave has responded with NAK (NAK is encoded as E0h in SMI-protocol-standard)
70h	internal error during communication (should not occur)
7Dh	SMI-bus without function due to missing supply power
7Eh	Communication error on the SMI-bus (slave-response could not be decoded)
7Fh	SMI-communication still uninitialized (may occur if a telegram transmission was requested too soon after an actuator reset)

Property 65h

The property 65h is read-only and is continuously updated with the status of the channels 1 through 8. Channels are assigned to bytes in the property from left (byte index 0) to right (byte index 7) as channel 1 through channel 8 respectively. The meaning of the byte-codes is as given in the following table:

Value	Meaning
00h	drive present, without error and moving in UP-direction
01h	drive present, without error and moving in DOWN-direction
02h	drive present, without error and at rest
03h	drive present, without error and motion state unknown by now
04h	drive error
08h	error due to SMI still not initialized
10h	error due to drive response NAK
20h	error in SMI-communication

40h	error due to missing SMI-supply power
8Fh	no drive assigned to this channel
9Fh	the drive that should be assigned to this channel, was not found
AFh	no programmable drive could be found for this channel
FFh	channel has not yet been checked

Generally the bits related to error codes 04h through 40h can be signaled in combination (the nibble then contains the respective codes added, i.e. logically ANDed).

Property 80

The property with ID 80 is read-only and provides a bit-mask of all detected SMI-slaves in the form that would be used for a group-addressing following SMI-protocol-standard, would all present drives be addressed in a single telegram.

Properties 81 through 88

The eight properties with the IDs 81 through 88 are read-only and reflect details about the SMI-slaves present (detected) on the SMI, ordered into the properties according to the assignment of the slaves to channels.

In the eight bytes of each of these properties the following information is coded ordered from left to right (i.e. in byte-indices from 0 to 7):

Byte Index	Bedeutung
0	number of the channel, the drive has been assigned to during initialization
1 high-nibble	drive-type-ID following SMI-standard (manufacturer specific)
1 low-nibble	Manufacturer-ID following SMI-Standard
2 ... 5	key-ID of the drive following SMI-standard (read from left to right, i.e. most significant byte in the position of the least byte-index / manufacturer specific)
6 ... 7	value of the relative SMI-position difference, determined during calibration; if the channel has not yet been calibrated but has been configured to be a venetian blind, the value will be FFFFh, for a drive configured to be a simple blind it will be 0000h

The SMI-Tool-Software

Access to the properties in the UIO 50001 by use of the device editor from e.g. the ETS3-package is somehow awkward and may be considerably tedious. It furthermore requires specific knowledge – in particular a thorough comprehension of the UIO-properties and their meanings – and some advanced experience of the user, e.g. being trained in decoding hexadecimal number notations. Especially if SMI-telegrams are to be transmitted to slaves via the gateway-property 64, they first have to be assembled manually and entered byte-wise in hexadecimal notation.

Moreover, a continuous monitoring of changing property-values as e.g. the state bytes in property 65, is not reasonably possible, since these may be updated rapidly by the actuator according to current situation. The device editor only allows for a manually triggered re-reading of a complete object, which may take quite a time in relation.

If installed, the device editor can always be used for reading but requires an additional licensing for writing if it has not been purchased already in connection with a manufacturer license. This is a next point is, if writing properties is demanded, that calls for some more appropriate utility. The SMI-tool aims at remedying these shortcomings.

The SMI-tool provides a simple diagnostic aid with convenient options and reasonable capabilities. By its use the access to actuators and the properties of the UIO 50001 is considerably facilitated.

Once the list of physical addresses of the actuators connected to the projected KNX-bus has been imported into the tool from a simple ASCII-file, one specific actuator and its connected SMI may easily be selected. The list may be exported from the ETS and/or be edited manually with a common ASCII-editor.

After selection of the actuator the read-only values of the properties 65, 80 and 81 through 88 are displayed clearly in a table. The values of the single bytes of property 65 reflecting all current drive states are frequently updated and visualized clearly assigned to the respective drive IDs.

The Use of the gateway-property 64 is supported in a reasonable way. Arbitrary (even manufacturer or drive-type specific configuration-) commands can still be entered in hexadecimal notation in an input-field and the reaction be monitored continuously, while the most common commands like move-UP/DOWN or stop can be issued simply by selecting the respective channel and pressing the according button with a mouse click. Last not least the identification of drives with key-IDs previously unknown is a straight forward matter, as the key-IDs detected are displayed as assigned to their channels.

The commands effecting special reactions of the actuator by use of the control-byte in the gateway-property can as well be issued by a simple mouse click on one of the according buttons.

Another enhancement of features has been added to the SMI-tool as a reaction to requests of users: an additional form supports a key-ID scan of the SMI-bus of the respective actuator that is controlled by the SMI-tool but executed solely by use of the actuator gateway. If invoked the tool applies the same search and addressing algorithm from the SMI-standard the SMIdrive-8 actuator employs for its scan when applying automatic addressing mode during first initialization. Up to eight of the slaves found may be programmed with addresses in this way, slave-address 8 being assigned the drive with the greatest key-ID, address 1 the one with the least.

Since the actuator itself is involved in this process only with its gateway function, it is suggested to have it perform a re-initialization afterwards in order to make sure, data of the connected slaves will be updated accordingly inside the actuator flash-memory.

The SMI-tool can be downloaded from the RK-AHT-website www.rk-aht.de. Further enhancements of functionality will be implemented as customer demands suggest.