ekinex

CONTROL YOUR LIVING SPACE

Application manual

KNX 6-channel universal interface EK-CB2-TP

Contents

1 Scope of the document

This application manual describes application details for the A1.0 release of the ekinex® KNX universal 6-channel interface EK-CB2-TP.

The document is aimed at the system configurator as a description and reference of device features and application programming. For installation, mechanical and electrical details of the device please refer to the technical description datasheet.

Application manual and application programs for ETS are available for download at www.ekinex.com.

Item	File name (## = release)	Version	Device rel.	Update
Technical datasheet	STEKCB2TP_EN.pdf		A1.0	03 / 2014
Application manual	MAEKCB2TP_EN.pdf	-		
Application program	APEKCB2TP##.knxprod	-		

You can access the most up-to-date version of the full documentation for the device using following QR code:

2 Product description

The ekinex® EK-CB2-TP is an S-mode KNX device that allows to connect switches and sensors of conventional type (not communicating natively on the KNX bus), equipped with potential-free contacts, to the KNX bus.

Its dual-purpose interface lines can also alternatively be used as output lines to drive low-power LEDs as indicators.

Through this universal module it is possible to employ normal switches, pushbuttons and sensors or binary signals made available by other devices to switch and control KNX bus functions.

The device inputs can be used either for independent single channels, e.g. for the connection of conventional switches or pushbuttons dedicated to the on/off switching of loads, or for coupled channels, e.g. for the connection of double pushbuttons for the control of dimmer or motorized drives.

Any number of the available lines can alternatively be configured as LED outputs, allowing for flexibility in applications; a line configured as output cannot act as input and vice-versa.

From here on, for the sake of brevity, the input / output lines will simply be referred to as “inputs” whenever there is no chance of misunderstanding.

The device is equipped with an integrated bus communication module and is designed for in-box mounting in combination with existing or new conventional switching points or devices.

In the input function, the device basically receives an input signal and converts it into a corresponding telegram sent on the bus; the telegram sent by the device is received and processed by one or more KNX actuators. In the output function, attached LEDs can be driven either according to the status of the device, or through bus telegrams originating from other KNX devices.

The device is powered by the KNX bus line with a 30 VDC SELV voltage and does not require auxiliary power; all required operation voltages for the input channels are produced inside the device.

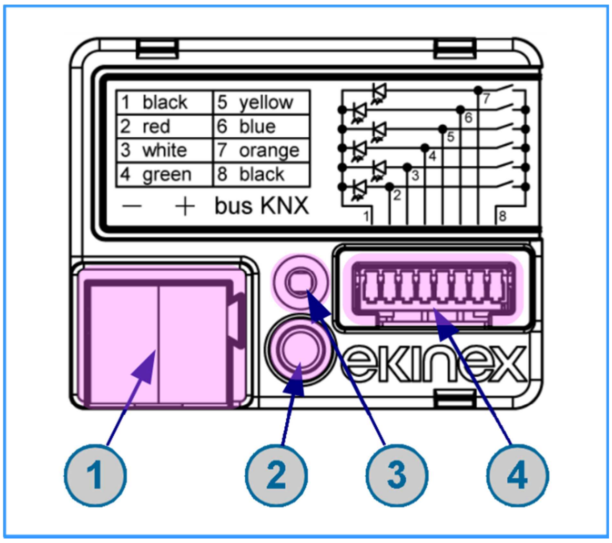
i

For further technical information, please also refer to the product datasheet STEKCB2TP_EN.pdf available on the ekinex website www.ekinex.com.

3 Switching, display and connection elements

The device is equipped with:

- a programming pushbutton and a programming LED
- a connector header for the input / output wires
- terminals for connecting the KNX bus line



1) Terminal block for KNX bus line
2) Programming pushbutton
3) Programming LED
4) connector header for the input / output wires

Fig. 1 - Switching, display and connection elements

Input signals are fed through the connector header; a matching plug with wires is supplied with the device.

The switch devices to be connected must be capable of supplying a potential-free contact, either Normally Open or Normally Closed. Voltage level signals (e.g. 24V signals) are not compatible with the device; a separation relay must be employed in case there is the need to interface such signals.

The wiring arrangement is detailed in Fig. 2, which is also drawn on the device label together with a reference to the corresponding wire colors.

The outer wires carry the same signal, which is the common line for both all inputs and all outputs.

Although the picture shows both connections possible for each line, i.e. as input and output, only one of these connections can be used at a time.

For further details, please refer to the technical datasheet of the device.

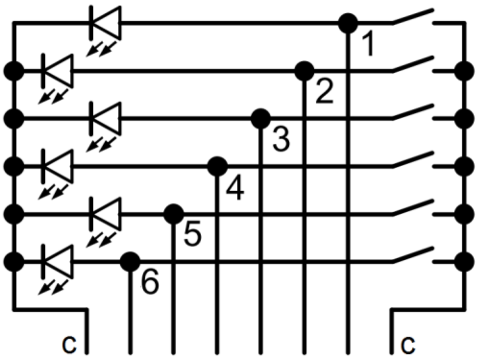


Fig. 2 - Input and output connections

4 Configuration

The exact functionality of the device depends on the software settings.

In order to configure and commission the device you need ETS4 or later releases and the ekinex® application program APEKCA1TP.knxprod which can be downloaded from the ekinex website www.ekinex.com.

The application program allows the configuration of all working parameters for the device.

The device-specific application program has to be loaded into ETS or, as alternative, the whole ekinex® product database can be loaded; at this point, all the instances of the selected device type can be added to the project.

For every single device, ETS allows to set the operating parameters individually for each input as described in detail in the following chapters.

The configuration can, and usually will, be performed completely offline; the actual transfer of the programmed configuration to the device takes place in the commissioning phase.

Product code	EAN	No. of channels	ETS application software (## = release)	Communication objects (max nr.)	Group adresses (max nr.)
EK-CB2-TP	8018417180965	6	APEKCB2TP##.knxprod	118	254

i

Configuration and commissioning of KNX devices require specialized skills. To acquire these skills, you should attend training courses at a training center certified by KNX.

For further information: www.knx.org

5 Commissioning

After the device has been configured within the ETS project according to user requirements, the commissioning of the device requires the following activities:

- electrically connect the device, as described in the product datasheet, to the bus line on the final network or through a purposely setup network for programming;
- apply power to the bus;
- switch the device operation to programming mode by pressing the programming pushbutton located on the rear side of the housing. In this mode of operation, the programming LED is turned on steady;
- upload the configuration (including the physical address) to the device with the ETS program.

At the end of the upload, the operation of the device automatically returns to normal mode; in this mode the programming LED is turned off. Now the device is programmed and ready for use on the bus.

6 Function description

After switching on the bus, which also acts as a power supply, the device becomes fully functional after a very short time needed for re-initialization. A delay is programmable for the device to become active on the bus in order to avoid a bus traffic overload during the first moments of startup of the whole network.

In case of a bus power failure (voltage lower than 19 V for 1 s or more), the device becomes unreactive: the timing functions are not active, neither are the programmed group addresses. As soon as the bus voltage is restored, the device will resume operation in its previous state (which is saved on power fail), unless different initialization settings are programmed.

6.1 Offline operation

A fully unprogrammed device causes no activity on the bus.

6.2 Online operation

In general the device works like a configurable digital sensor that is listening to own inputs or outputs of other devices. On input events the device performs output functionality over KNX bus like sending values or controlling external devices like KNX actuators.

6.2.1 Software working cycle

The main purpose of the software is to switch relays as the result of processing the information transmitted on a telegram received from the KNX bus. The software working cycle can be described as follows:

- Handle input contacts or user pushbutton presses and generate bus telegrams according to the assigned functions;
- Implement input / pushbutton interlock and timing functions;
- Handle incoming bus messages in order to update the status of pushbutton activations and LED indicators;
- Respond to bus messages requesting feedback on the status of the inputs.

There are also special events on which it is possible to trigger additional features. These events are the bus failure and recovery, and the download of a new configuration with ETS. Triggered features can be device relay switching and activation of supplementary functions. The output value result depends on the device state and parameters.

6.2.2 Inputs

6.2.2.1 Input types

The digital sensor triggers its functionality on events on digital inputs; digital values on the input correspond to the status of connected physical contacts.

The device may be configured in two modes, so as to be interfaced to different contacts: **NO** (normally open) and **NC** (normally closed).

Usually, the mode denomination matches the type of the contact of the electrical device used at the input. From a logical point of view, this mode affects the interpretation of the "active" and "inactive" state of an input as follows:

- In NO mode, an open connection between the terminals (open contact) is associated to the inactive state, while a closed contact is associated to the active state;

- in NC mode, an open connection between the terminals (open contact) is associated to the active state, while a closed contact is associated to the inactive state.

6.2.2.2 Input events

The device recognizes two types of input events: "*close / open contact*" and "*short / long press*".

The first event type is a simple logical value change: "OPEN" is an alias for "inactive", whereas "CLOSE" means "active state".

> *It is very important to stress that the words "OPEN" and "CLOSE", although standard terms for input status conditions, are to be interpreted from a logic point of view, and that they are not to be confused with the physical contact status as used in the description of "NO" and "NC" input types.*
>
> *In other words, for example, a NO contact in active position is electrically AND logically CLOSEd, whereas a NC contact in active position is electrically open, but logically CLOSEd.*

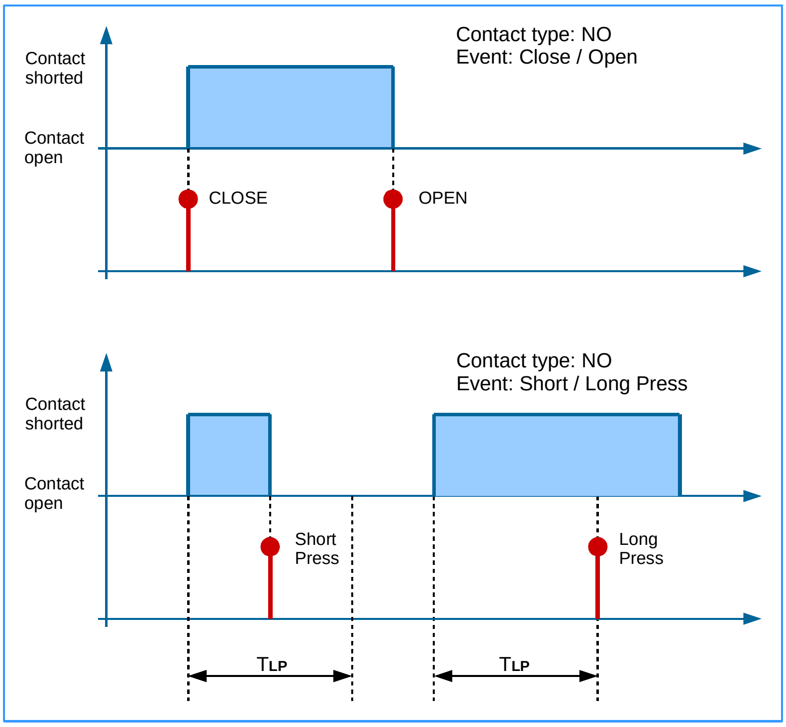
The second type of event that can be associated with an input is the Short or Long Press; the term "press" is typically referred to user activated pushbuttons, although it also applies to signals originating from contacts of other devices.

The distinction is as follows:

- If an input remains active for a period shorter than a defined time duration, upon release a "Short Press" event is generated;
- If the input remains active for longer than the defined time duration, at the duration time point a "Long Press" event is generated. Thereafter, the input can remain active for as long as desired, and no more events are generated either during the rest of activation or at release (next event will occur after next activation).

Please refer to time diagrams in following figures for an illustration of the difference between these events.

Time diagrams for normally open (NO) mode:



Time diagrams for normally close (NC) mode:

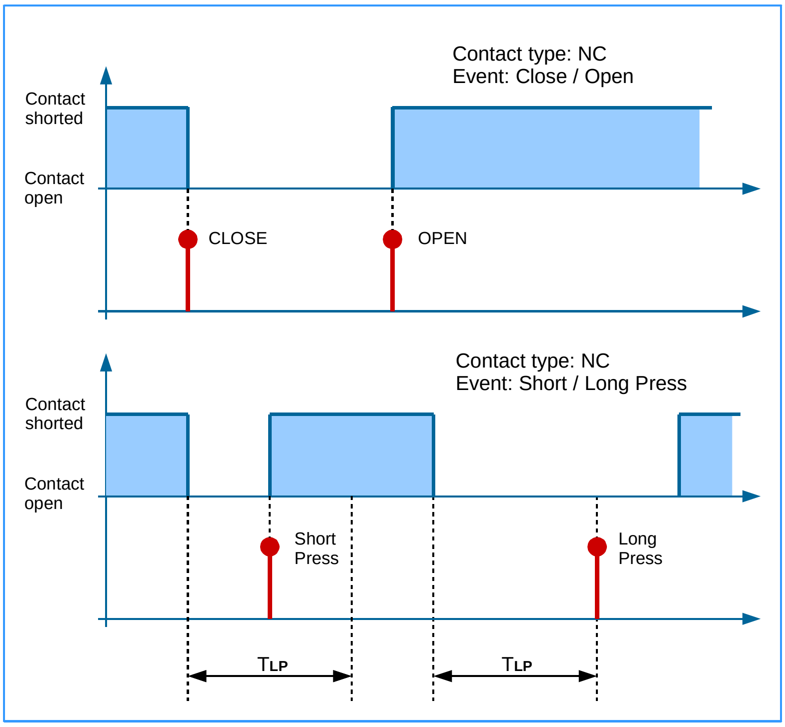


Figure 2: time diagrams for NO and NC modes

6.2.2.3 *Lock function*

For each input channel (or input pair, see below), a lock feature can be enabled which allows to block the operation of an input channel by changing a value of a communication object.

When in a locked state, the input channel is effectively disabled; the locked state can be deactivated by sending another telegram.

A value (for each transition) can be specified to be assigned to the communication object upon entering or exiting the locked state.

The locked state can also be automatically activated when the bus is connected.

6.2.3 State variables

6.2.3.1 *State variables (Communication objects)*

The variables that are changed by the input events can be one of the types available for KNX communication objects, i.e. for instance a 1 bit value (on-off), a 2 bit value or an integer value of larger size.

In all cases, each of the two events can:

- change the value of the variable to one of two definable values within its range (which is trivial in the case of the 1-bit value);
- toggle between the two defined values
- do nothing (value is unaffected)

This state variable, once assigned a group address, is actually a **KNX communication object**; as such, it undergoes the usual rules for communication objects, among which – for instance – the effect of flags to determine how the change of value affects the transmission of the objects.

This obviously implies that, if the value of a communication objects changes due to the effect of a bus telegram, the corresponding change will register in the device, according to its associated flags.

6.2.3.2 Binding between Events and Communication objects

The above description is a little simplified in order to ease comprehension; as a matter of fact, to each event can be assigned not just one, but several communication objects (up to 8), even of different types. Each of these communication objects can have its own behavior and its own associated value set.

6.2.3.3 Repeated send

For most features, is it possible to set the device to send a telegram not just when a value changes as a consequence of an input transition, but also at regular intervals whenever that value setting is active.

This behavior, also referred to as Cyclical Transmission, can be set separately for each of the two values that are associated to an input (or both, or none of them).

If an input is set to "*send values or sequences*" mode, repeated send is not available if more than 1 Communication Object is assigned to that input.

6.2.4 Input coupling

The 6 input channels of the device can be considered, and used, as independent; however, due to the physical structure of the device and the nature of the functions it most frequently performs, these inputs can be naturally grouped in pairs. In this case, each channel is made of a pair of inputs which are physically close on the terminal block.

In order to specify channel pairings, each input can be configured in two ways: single mode and coupled mode.

- In *single mode*, each input operates independently, has its own parameters and communication objects. This is the mode of operation described so far.
- In *coupled mode*, 2 inputs operate logically grouped under a channel in order to perform a common functionality. Only adjacent inputs belonging to the same channel can be coupled, therefore the only combinations allowed for coupling are 1 with 2, 3 with 4, and 5 with 6.

It is possible to configure some of the inputs in single mode and the others in coupled mode, with the pairing constraints just described.

Single and coupled modes have a similar functionality, but differ for the configuration.

6.2.4.1 Single input mode

Each single input can be configured for one of following different features:

1. *Send values or sequences*
   An event triggers the transmission on the bus of configurable values or sequence of values. These values can be of a logical type or a numerical type with a different size.

   A sequence of values can be made of up to 8 communication objects of different value types; time delays can set between values in the sequence.
2. *Dimmer control*
   This mode is intended to be used with dimming actuators for the control of lighting devices.

   The functionality is triggered on short press and long press events. On short press events, the device sends on/off telegrams to the dimming actuator; on long press events, the dimming percentage is varied up or down until the button is released.

3. *Shutter or Venetian blind control*
   This mode is intended to be used together with actuators for the control of motorized blinds, shutters and similar devices. These actuators have functions for blind opening and closing; two movement types are selectable, i.e. continuous movement and stepwise movement. On input events, the device sends operation telegrams to the actuators.

   The operation is configurable through following parameters:

   - If *toggle* mode is enabled, on each activation of the same input the movement direction is inverted; if it is disabled, the movement direction is fixed and it can be set to "up" or "down".
   - If *blinds* mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.

4. *Scene function output*
   This mode is intended to be used together with several KNX actuators that support using a scene function; this function allows to store and recall a communication object value on an actuator.

   In this mode, the role of the device is to send a "store / recall scene" telegram to the actuator on a long / short press event.

   This mode has two possible configurations:

   - Activate preset scene on short press, and store current setting as scene value on long press
   - Activate two different scenes on long and short press.

5. *Pulse counter*
   In this mode the device could count the number of commutations at an input channel. The counter value can be read from a communication object which can be cyclically sent on specified time period. It is possible to set the counter's value type and maximum reachable value.

6.2.4.2 Coupled Input mode

Each pair of coupled inputs can be configured for one of following different features (only the differences from the single mode are highlighted):

1. *Switch control*
   Both inputs in a pair are bound to the same communication object; unlike single mode, the object can only be of the 1-bit type (on-off), therefore building a conventional switching behaviour.

   The user can configure which of the two inputs sets the "off" or resp. "on" value.

2. *Dimmer control*
   The functionality is triggered on short press and long press events of the inputs in the pair.

   The user can configure which of the two inputs sets the "up" or resp. "down" value.

   On short press events, the input configured as "up" sends an "on" switching telegram to the dimming actuator; while the "down" input sends an "off" telegram.

   On long press events, the dimming percentage is varied up or down until the button is released.

3. *Shutter or Venetian blind control*
   The two inputs of a pair are assigned to opposite movement directions; these can be assigned to inputs as desired, i.e. A up / B down or the other way around.

   The *blinds* mode can also be set, and it works exactly as in single mode.

In coupled mode there is no provision either for a *scene* control feature or for a *counter* feature.

6.2.5 Outputs

Any of the interface lines can be used to connect an indicator LED, with an exception: an interface line can be used as LED output only if it's not part of a channel, i.e. of an input pair.

LEDs can be driven through KNX telegrams in one of following ways:

- Fixed value (always on or always off);
- Status set from the bus through a communication object. In this case, the LED can be set to be flashing when active (with a choice of different on/off time combinations), and the on/off light status can be inverted with respect to the communication object status (so as to have the LED on when the CO has an "off" value).

6.3 Device settings

This section lists all configurable parameters and describes related communication objects.

Every channel, and every input or input pair under a channel, offers the same set of communication objects and parameters, but they may all be independently configured.

Hereafter, all channel-specific settings are listed grouped by channel; a generic channel number is referenced as "x" (where x = 1…6).

> i The parameter values highlighted in bold represent the default value.

The device settings are divided in two main groups: the *general* settings and the *channel-specific* settings.

6.3.1 General settings

The parameters in this section define the overall behavior of the device.

Parameter name	*Conditions*	*Settings*
Debounce time	-	hh:mm:ss.fff (**00:00:00.020**)
	Sets a minimum time during which an input must remain stable in order to be considered valid, in order to avoid contact bounces or spikes .	
Delay after bus voltage recovery	-	hh:mm:ss.fff (**00:00:04.000**)
	Delay before bus telegrams can be sent after a recovery of the bus voltage. The delay time affects the transmission generated by an event as well as the cyclical transmission. For the cyclical transmission: after the delay time finished, the cycle restarts and the first telegram will be sent after the cycle time.	

6.3.2 Channels configuration

These settings configure device channel behavior.

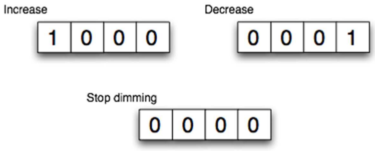
Parameter name	*Conditions*	*Settings*
Channels 1 and 2 Channels 3 and 4 Channels 5 and 6	-	disabled **independent** coupled copy parameters from channels*
	Set operation mode for inputs. * *This option is only available for channel pairs 3/4 and 5/6. If selected, the corresponding channel pair can be made to perform the exact same kind of function the specified pair, but <u>basing on different communication objects</u>.* *This allows to spare time in configuring the device, at the same time assuring that there is no inconsistency between channels that are meant to be configured in exactly the same way.* *To assign the same configuration is just a shortcut for the selection of configuration options; it is in no way implied that the channels share any of the involved communication objects (each channel has its own independent objects).*	
Channel to copy from	**Channels *x* and *y* = copy parameters from channel** (x > 1)	**1 and 2** 3 and 4*
	All the parameters of the specified channel pair are copied from the chosen source channel. * *Only selectable for pair 5/6.*	
Channel *x*	**Channel *x* = independent**	disabled **input** led output copy parameters from channel*
	Set direction for channel or disables it. * *See previous notes.*	
Type	**Channels *x* and *y* = independent** **Channel *x* = input**	**send values or sequences** dimming shutter or venetian blind scene counter
	Determines the kind of function performed by the specified input. *Further parameters for the selected function will appear in the individual input configuration sections (see below).*	
Type	**Channel *x* = coupled**	switching **dimming** shutter or venetian blind
	Determines the kind of function performed by the input pair. *Further parameters for the selected function will appear in the individual rocker configuration sections (see below).*	

6.3.2.1 *Independent: send values or sequences*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *x* – Switching status *[type]*, object *n*	Channels *x* and *y* = independent Channel *x* = input Type = send values or sequences	*According to configuration* (**1-bit**)	CRWTU	*According to configuration* (**[1.001] switch**)	5, 22 (1, 2) 43, 60 (3, 4) 81, 98 (5, 6)
	Up to 8 objects can be defined for binding with the same event. *The listed CO numbers are those referring to object nr.1; the COs for each subsequent object are following in sequence.* *To obtain the CO numbers for object number n, just add (n-1) to the listed numbers.* *E.g.: COs associated to input 5 have numbers from 81 to 89. The number of CO nr. 3 is therefore 81+(3-1) = 83.* ***The size and type of the individual objects can be configured as described in following sections****.*				

6.3.2.2 *Independent: dimming*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *x* – Switching command	Channels *x* and *y* = independent Channel *x* = input Type = dimming	1 bit	CR-T-	[1.001] switch	13, 30 (1, 2) 51, 68 (3, 4) 89, 106 (5, 6)
	Send a command to a dimming actuator to switch the light on or off. *The command is triggered by a short press on the input.* *The value sent can be a fixed value or it can be toggled at each input activation.*				
Input *x* – Dimming up / down / stop command	Channels *x* and *y* = independent Channel *x* = input Type = dimming	4 bit	CR-T-	[3.*] 3-bit control	14, 31 (1, 2) 52, 69 (3, 4) 90, 107 (5, 6)
	Send a command to a dimming actuator to change dimming intensity (brighter or darker). *Three values are used which mean start increase, start decrease or stop the change.* Increase: 1 0 0 0 — Decrease: 0 0 0 1 — Stop dimming: 0 0 0 0 *Increase/decrease values are sent when a long press action occurs and stop value on press release.* *The value sent can be a fixed value or it can be toggled at each input activation.*				

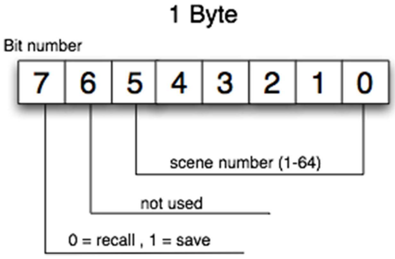


6.3.2.3 *Independent: shutter or venetian blind*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *x* – Dedicated stop command	Channels *x* and *y* = independent Channel *x* = input Type = shutter or venetian blind	1 bit	CR-T-	[1.017] trigger	13, 30 (1, 2) 51, 68 (3, 4) 89, 106 (5, 6)
	Immediately stop any movement of the blind. The object is sent on a short press if the blind mode is disabled, and at the end of a long press if the venetian blind mode is enabled.				
Input *x* – Stop – step up/down command	Channels *x* and *y* = independent Channel *x* = input Type = shutter or venetian blind Venetian Blind mode = enabled*	1 bit	CR-T-	[1.007] step	16, 33 (1, 2) 54, 71 (3, 4) 92, 109 (5, 6)
	Move the blind to fully open or fully closed position. The object is sent at the end of a long press. ** This setting is configurable under a further menu for the individual channel, described below; it is listed here for convenience.*				

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *x* – Move up / down command	Channels *x* and *y* = independent Channel *x* = input Type = shutter or venetian blind Venetian Blind mode = disabled*	1 bit	`CR-T-`	[1.008] up/down	17, 34 (1, 2) 55, 72 (3, 4) 93, 110 (5, 6)
	Increase or decrease the opening of the blind stepwise. *The object is sent on a short press.* ** This setting is configurable under a further menu for the individual channel, described below; it is listed here for convenience.*				

6.3.2.4 *Independent:scene*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *x* – Scene number	Channels *x* and *y* = independent Channel *x* = input Type = scene	1 Byte	`CR-T-`	[17.*] Scene number [18.*] Scene control	18, 35 (1, 2) 61, 73 (3, 4) 94, 111 (5, 6)
	Store or recall a scene. The lowest 6 bits in the byte form the code of the scene, while the highest bit is the operation code (store or recall).  1 Byte Bit number 7 6 5 4 3 2 1 0 scene number (1-64) not used 0 = recall , 1 = save				

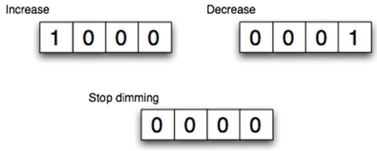
6.3.2.5 *Independent: counter*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input x – Counter value	Channels *x* and *y* = independent Channel *x* = input Type = counter	*According to configuration* (**1-bit**)	`CR-T-`	[12.001] Counter pulses [13.001] Counter pulses	18, 35 (1, 2) 61, 73 (3, 4) 94, 111 (5, 6)
	This object stores the current value of the input counter				
Input x – Counter reset command	Channels *x* and *y* = independent Channel *x* = input Type = counter	1-bit	`C-W--`	[1.015] reset	19, 36 (1, 2) 62, 74 (3, 4) 95, 112 (5, 6)
	Reset the counter by setting its value to zero				
Input x – Counter runout	Channels *x* and *y* = independent Channel *x* = input Type = counter	1-bit	`CR-T-`	[1.055] alarm	20, 37 (1, 2) 63, 75 (3, 4) 96, 113 (5, 6)
	Send an alarm bit when the counter reaches the maximum value according to the data size defined for the counter.				

6.3.2.6 *Coupled: switching*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Inputs *x* and *y* – Switching command	Channels *x* and *y* = coupled **Type = switching**	1-bit	CRWTU	[1.001] switch	13 51 89

6.3.2.7 *Coupled: dimming*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Inputs *x* and *y* – Switching command	Channels *x* and *y* = coupled **Type = dimming**	1 bit	CRWTU	[1.001] switch	13 51 89
	Send a command to a dimming actuator to switch the light on or off. *The command is triggered by a short press on the input.* *The value sent can be a fixed value or it can be toggled at each input activation.*				
Inputs *x* and *y* – Dimming up / down / stop command	Channels *x* and *y* = coupled **Type = dimming**	4 bit	CR-T-	[3.*] 3-bit control	14 52 90
	Send a command to a dimming actuator to change dimming intensity (brighter or darker). *Three values are used which mean start increase, start decrease or stop the change.*  Increase 1 0 0 0 Decrease 0 0 0 1 Stop dimming 0 0 0 0 *Increase/decrease values are sent when a long press action occurs and stop value on press release.* *The value sent can be a fixed value or it can be toggled at each input activation.*				

6.3.2.8 *Coupled: shutter or venetian blind*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Inputs *x* and *y* – Dedicated stop command	Channels *x* and *y* = coupled **Type = shutter or venetian blind** **Venetian Blind mode = disabled***	1 bit	CRWTU	[1.017] trigger	13 51 89
	See same option in independent mode for details				
Inputs *x* and *y* – Stop – step up/down command	Channels *x* and *y* = coupled **Type = shutter or venetian blind** **Venetian Blind mode = enabled***	1 bit	CR-T-	[1.007] step	16 54 92
	See same option in independent mode for details				
Inputs *x* and *y* – Move up / down command	Channels *x* and *y* = coupled **Type = shutter or venetian blind**	1 bit	CRWTU	[1.008] up/down	17 55 93
	See same option in independent mode for details				

6.3.3 Channel *x* and *y:* Input *x* and *y* configuration

6.3.3.1 Independent channels

For the *independent or* single channel setting, all parameters listed below are referred to either input.

In the following sections, it is implicitly understood that for the listed parameters to appear, the corresponding inputs must be enabled.

The entries assigned to "Object *n*" are repeated so many times as the number of configured objects according to the *Number of Communication Objects* parameter.

For all Type values:

Parameter name	*Conditions*	*Settings*
Contact type	-	**NO (normally open)** NC (normally closed)
	In normally open (NO) mode the "active" state of the input is when input contacts shorted, and the "inactive" state is when they are disconnected. In normally closed (NC) mode the device behaviour is the opposite.	
Lock function	-	enabled / **disabled**
	Enables or disables the capability of locking the input through a remote command (telegram).	
Lock function – Invert lock device signal	Channels *x* and *y* = independent Type = send values or sequences	**not inverted** / inverted
	Allows to interpret a "lock activate" telegram as unlock and vice-versa.	
Lock function – Lock after bus recovery	Channels *x* and *y* = independent Type = send values or sequences	**no** / yes
	If active, after returning from a bus failure or power-off the device will retain the lock status it had before. Otherwise (in the default case), the device will restart in the non-locked condition.	

6.3.3.2 Independent: Lock function enabled

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Inputs *x* and *y* – Lock command	Channels *x* and *y* = independent Lock function = enabled	1 bit	`C-W--`	[1.003] enable	4,21 (1, 2) 42,59 (3, 4) 80,97 (5, 6)

When the lock function is enabled, for each input or channel a behaviour can be defined to be followed when the locking or unlocking command is received.

The details will be listed in the following sections; the different behaviours are summarized in the table below.

Channel mode	*Input type*	*Behaviour at locking*	*Behaviour at unlocking*
independent	send values or sequences	**none** as close or short press as open or long press	
	switching	**none** off on toggle	**none** off on as previous
coupled			
independent	dimming		
coupled			
independent	shutter or venetian blind	none up down	
coupled			
independent	scene	none send first scene send second scene	
independent	counter	none send counter value	

6.3.3.3 Independent: send values or sequences

Parameter name	*Conditions*	*Settings*
Number of Communication Objects	Channels *x* and *y* = independent Type = send values or sequences	1…8 (**1**)
	Number of configuration objects configured in association with the input event.	
Lock function – Behaviour at locking	Channels *x* and *y* = independent Type = send values or sequences	**none** as close or short press as open or long press
	Allows to perform the operation associated to the specified event when a locking command is received.	
Lock function – Behaviour at unlocking	Channels *x* and *y* = independent Type = send values or sequences	**none** as close or short press as open or long press
	Allows to perform the operation associated to the specified event when an unlocking command is received.	
Event	Channels *x* and *y* = independent Type = send values or sequences	**close / open contact** short / long press
	Type of event that should be used as trigger for an action	
Long press time	Channels *x* and *y* = independent Type = send values or sequences Event = short /long press	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Object *n* – send delay	Channels *x* and *y* = independent Type = send values or sequences	hh:mm:ss.ff (**00:00:00.00**)
	Delay before the object is transmitted on the bus. *By defining a delay after the event occurs and before the object value is sent, it is possible to associate a time defined sequence of values to an input event.*	
Object *n* – Cyclical transmission	Channels *x* and *y* = independent Type = send values or sequences **Number of Comm. Objects = 1**	none **off / value 1** on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated. *Cyclical transmission is only available if the number of Communication Objects is set to “1”.*	
Object *n* – Cyclical transmission interval	Channels *x* and *y* = independent Type = send values or sequences **Number of Comm. Objects = 1** **Send cyclically ≠ none**	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Object *n* – Communication Object dimension	Channels *x* and *y* = independent Type = send values or sequences	**1 bit value** 2 bits value 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned value 2 bytes signed value 2 bytes floating point value
	Defines size and type of the values to be sent when an event occurs.	

Parameter name	*Conditions*	*Settings*
Object *n* – Close *or* Short Press	Channels *x* and *y* = independent Type = send values or sequences **send dimension = 1 bit value**	none on off **toggle**
	Channels *x* and *y* = independent Type = send values or sequences **send dimension = 2 bit value**	none disable enable off / up **enable on / down** enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Channels *x* and *y* = independent Type = send values or sequences **send dimension = *any byte value***	none **send value 1** send value 2 send value 1 ↔ send value 2
	Value change behaviour caused by either a Close or a Short Press event (according to event configuration)	
Object *n* – Open *or* Long Press	Channels *x* and *y* = independent Type = send values or sequences **send dimension = 1 bit value**	**none** on off toggle
	Channels *x* and *y* = independent Type = send values or sequences **send dimension = 2 bit value**	none **disable** enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Channels *x* and *y* = independent Type = send values or sequences **send dimension = *any byte value***	**none** send value 1 send value 2 send value 1 ↔ send value 2
	Value change behaviour caused by either an Open or a Long Press event (according to event configuration)	
Object *n* – Value 1	Channels *x* and *y* = independent Type = send values or sequences **send dimension = *any byte value***	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 bytes unsigned value) -32768... 32767 (2 bytes signed value) -671088.64...670760.96 (2 bytes floating value)
	First value available for association in send events	
Object *n* - Value 2	Channels *x* and *y* = independent Type = send values or sequences **send dimension = *any byte value***	*same as value 1*
	Second value available for association in send events	

Parameter name	*Conditions*	*Settings*
Object *n* - Value sent after bus on	Channels *x* and *y* = independent Type = send values or sequences **send dimension = 1 bit value**	**none** on off previous
	Channels *x* and *y* = independent Type = send values or sequences **send dimension = 2 bit value**	none **disable** enable off / up enable on / down previous
	Channels *x* and *y* = independent Type = send values or sequences **send dimension = *any byte value***	**none** send value 1 send value 2 previous
	Value to be sent after a recovery of the bus voltage.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Inputs *x* and *y* – Switching status *[type]* Object *n*	Channels *x* and *y* = independent Type = send values or sequences	*See table below*	`CRWTU`	*See table below*	5, 22 (1, 2) 43, 60 (3, 4) 81, 98 (5, 6)
	The listed CO numbers are those referring to object nr.1; the COs for each subsequent object are following in sequence. *To obtain the CO numbers for object number n, just add (n-1) to the listed numbers.* *E.g.: COs associated to input 3A (of Channel 3) have numbers from 81 to 89. The number of CO nr. 5 is therefore 81+(5-1) = 85.*				

Sizes and DPTs are as follows:

Size	*DPT*
1 bit	[1.001] switch
2 bits	[2.*] 1-bit controlled
1 byte unsigned	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte percentage	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte signed	[6.*] 8-bit signed value
2 bytes unsigned	[7.*] 2-byte unsigned value
2 bytes signed	[8.*] 2-byte signed value
2 bytes floating	[9.*] 2-byte float value

6.3.3.4 Independent: dimming

Parameter name	*Conditions*	*Settings*
Long press time	Channels *x* and *y* = independent Type = dimming	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Toggle mode	Channels *x* and *y* = independent Type = dimming	**enabled** / disabled
	When enabled, causes the short press to toggle the on-off status of the destination CO; otherwise, a fixed status can be assigned to the short press.	
Long press	Channels *x* and *y* = independent Type = dimming **Toggle mode = enabled**	**darker** brighter darker ↔ brighter
	Defines the function to be assigned to the long press. If the toggle mode is enabled, the Short press action is already defined as toggle.	
Short / Long action	Channels *x* and *y* = independent Type = dimming **Toggle mode = disabled**	**off / darker** on / brighter off / darker ↔ brighter on / darker ↔ brighter
	Defines the function to be assigned to the long and short press.	
Cyclical transmission	Channels *x* and *y* = independent Type = dimming	**none** off / value 1 on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclical transmission interval	Channels *x* and *y* = independent Type = dimming **Send cyclically ≠ none**	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at unlocking	Channels *x* and *y* = independent Type = dimming	**none** off on as previous
	Operation to perform when an unlocking command is received.	
Lock function – Behaviour at locking	Channels *x* and *y* = independent Type = dimming	**none** off on toggle
	Operation to perform when a locking command is received.	
Value sent after bus on	Channels *x* and *y* = independent Type = dimming	**none** off on previous
	Value to be sent after a recovery of the bus voltage.	

6.3.3.5 Independent: shutter or venetian blind

Parameter name	*Conditions*	*Settings*
Long press time	Channels *x* and *y* = independent Type = shutter or venetian blind	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Toggle mode	Channels *x* and *y* = independent Type = shutter or venetian blind	**enabled** / disabled
	When enabled, causes each subsequent press to invert the direction of movement; otherwise, a fixed direction can be assigned.	
Up / Down action	Channels *x* and *y* = independent Type = shutter or venetian blind **Toggle mode = disabled**	**up** down
	Defines the movement direction to be assigned to the button press.	
Venetian blind mode	Channels *x* and *y* = independent Type = shutter or venetian blind	enabled / **disabled**
	If blinds mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.	
Lock function – Behaviour at locking	Channels *x* and *y* = independent Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when a locking command is received.	
Lock function – Behaviour at unlocking	Channels *x* and *y* = independent Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when an unlocking command is received.	
Value sent after bus on	Channels *x* and *y* = independent Type = shutter or venetian blind	**none** up down
	Value to be sent after a recovery of the bus voltage.	

6.3.3.6 Independent: scene

Parameter name	*Conditions*	*Settings*
First scene number	Channels *x* and *y* = independent Type = scene	1…64 (**1**)
	Main scene number to be assigned to button press. It is named "first" for the case that an alternative scene number is used.	
Learning mode	Channels x and y = independent Type = scene	enabled / **disabled**
	When enabled, a long key press can be used to program the selected scene by storing the current parameters.	
Long press time	Channels x and y = independent Type = scene **Learning mode = enabled**	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Scene activation	Channels x and y = independent Type = scene **Learning mode = disabled**	**send first scene only** toggle between two scenes
	Allows the key to be used to alternate between two different scenes.	
Second scene number	Channels x and y = independent Type = scene **Learning mode = disabled** **Scene activation = toggle between two scenes**	1…64 (**2**)
	Alternate scene number to be assigned to button press.	
Lock function – Behaviour at locking	Channels x and y = independent Type = scene	**none** send first scene send second scene
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Channels x and y = independent Type = scene	**none** send first scene send second scene
	Operation to perform when an unlocking command is received.	
Value sent after bus on	Channels x and y = independent Type = scene **Scene activation = send first scene only**	**none** first scene
	Channels x and y = independent Type = scene **Scene activation = toggle between two scenes**	**none** first scene second scene previous
	Value to be sent after a recovery of the bus voltage.	

6.3.3.7 Independent: counter

Parameter name	*Conditions*	*Settings*
Cyclical transmission interval	Channels x and y = independent Type = counter	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions. A zero value (00:00:00) means no cyclical transmission.	

Counter dimension	Channels x and y = independent
	Type = counter
		from 0 to 255 (1 byte)
		from 0 to 65535 (2 bytes)
		from 0 to 4294967295 (4 bytes)

	Value type of the counter. Unsigned integer value of 1, 2 or 4 bytes.	
Max value	Channels x and y = independent Type = counter	Depending on the counter dimension: 0…**255** 0…**65535** 0…**4294967295** (default value is the maximum value of the selected interval)
	Limit value for the counter. When this value is reached,a “runout” telegram is sent and the counter value is reset to zero.	

6.3.3.8 Coupled

For a *coupled* channel, all the parameters are referred to the single menu entry for Input x and y.

In the following sections, it is implicitly understood that for the listed parameters to appear, the corresponding channel must be enabled.

For all Type values:

Parameter name	*Conditions*	*Settings*
Lock function	Channels x and y = coupled	enabled / **disabled**
	Enables or disables the capability of locking the input through a remote command (telegram).	

6.3.3.9 Coupled: Lock function enabled

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Inputs *x* and *y* – Lock command	Channels x and y = coupled **Lock function = enabled**	1 bit	C-W--	[1.003] enable	4 42 80

6.3.3.10 Coupled: switching

Parameter name	*Conditions*	*Settings*
x and *y* use	Channels x and y = coupled Type = switching	**input 1 on, input 2 off** input 1 off, input 2 on
Cyclical transmission	Channels x and y = coupled Type = switching	**none** off / value 1 on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclical transmission interval	Channels x and y = coupled Type = switching **Send cyclically ≠ none**	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at locking	Channels x and y = coupled Type = switching	**none** off on toggle
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Channels x and y = coupled Type = switching	**none** off on previous
	Operation to perform when an unlocking command is received.	
Value sent after bus on	Channels x and y = coupled Type = switching	**none** off on previous
	Value to be sent after a recovery of the bus voltage.	

6.3.3.11 Coupled: dimming

Parameter name	*Conditions*	*Settings*
Long press time	Channels x and y = coupled Type = dimming	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
x and *y* use	Channels x and y = coupled Type = dimming	**input 1 increases, input 2 decreases** input 1 decreases, input 2 increases
Cyclical transmission	Channels x and y = coupled Type = dimming	**none** off / value 1 on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclical transmission interval	Channels x and y = coupled Type = dimming **Send cyclically ≠ none**	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at locking	Channels x and y = coupled Type = dimming	**none** off on toggle
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Channels x and y = coupled Type = dimming	**none** off on toggle
	Operation to perform when an unlocking command is received.	
Value sent after bus on	Channels x and y = coupled Type = dimming	**none** off on previous
	Value to be sent after a recovery of the bus voltage.	

6.3.3.12 Coupled: shutter or venetian blind

Parameter name	*Conditions*	*Settings*
Long press time	Channels x and y = coupled Type = shutter or venetian blind	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
x and *y* use	Channels x and y = coupled Type = shutter or venetian blind	**input 1 up, input 2 down** input 1 down, input 2 up
Blind mode	Channels x and y = coupled Type = shutter or venetian blind	enabled / **disabled**
	If blinds mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.	
Lock function – Behaviour at locking	Channels x and y = coupled Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when a locking command is received.	
Lock function – Behaviour at unlocking	Channels x and y = coupled Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when an unlocking command is received.	
Value sent after bus on	Channels x and y = coupled Type = shutter or venetian blind	**none** up down
	Value to be sent after a recovery of the bus voltage.	

For other communication objects related to *coupled* mode, please refer to the general *Channels Configuration* section.

7 Appendix

7.1 Communication objects table

Following is a summary of all KNX Communication Objects (CO) and corresponding Data Point Types (DPT) defined by the application program according to configuration options.

The listing order is generally by CO number (in case of COs repeated by channel, the CO number for the first channel is taken as relevant).

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *x* / Inputs *x* and *y* – **Lock command**	Channels x and y = independent, Lock function = enabled	1 bit	`C-W--`	[1.003] enable	4, (1 / 1+2) 21, (2) 42, (3 / 3+4) 64, (4) 80, (5 / 5+6) 97 (6)
	Channels x and y = coupled Lock function = enabled				
Input *x* – **Switching status** *[type]* **Object *n****	Channels x and y = independent **Type = send values or sequences**	***See table 1***	`CRWTU`	***See table 1***	5..12, (1) 22..29, (2) 43..50, (3) 60..67, (4) 81…88, (5) 98..105 (6)
		** The listed CO numbers are starting from object nr.1; the COs for each subsequent object are following in sequence.* *To obtain the CO numbers for object number n, just add (n-1) to the listed numbers.E.g.: COs associated to input 5 have numbers from 81 to 88. The number of CO nr. 3 is therefore 81+(3-1) = 83.*			
Input *x* / Inputs *x* and *y* – **Switching command**	Channels x and y = independent, **Type = dimming**	1 bit	`CRWTU`	[1.001] switch	13, (1 / 1+2) 30, (2) 51, (3 / 3+4) 68, (4) 89, (5 / 5+6) 106 (6)
	Channels x and y = coupled **Type = switch**				
	Channels x and y = coupled **Type = dimming**				
Input *x* / Inputs *x* and *y* – **Dedicated stop command**	Channels x and y = independent **Type = shutter or venetian blind**	1 bit	`CRWTU`	[1.017] trigger	13, (1 / 1+2) 30, (2) 51, (3 / 3+4) 68, (4) 89, (5 / 5+6) 106 (6)
	Channels x and y = coupled **Type = shutter or venetian blind** **Blind mode = disabled**				

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *x* / Inputs *x* and *y* – **Dimming up / down / stop command**	Channels x and y = independent, **Type = dimming**	4 bit	`CR-T-`	[3.*] 3-bit control	14, (1 / 1+2) 31, (2) 52, (3 / 3+4) 69, (4) 90, (5 / 5+6) 107 (6)
	Channels x and y = coupled **Type = dimming**				
Input *x* / Inputs *x* and *y* – **Stop – step up/down command**	Channels x and y = independent, **Type = shutter or venetian blind** **Blind mode = enabled**	1 bit	`CR-T-`	[1.007] step	16, (1 / 1+2) 33, (2) 54, (3 / 3+4) 71, (4) 92, (5 / 5+6) 109 (6)
	Channels x and y = coupled **Type = shutter or venetian blind** **Blind mode = enabled**				
Input *x* / Inputs *x* and *y* – **Move up / down command**	Channels x and y = independent, **Type = shutter or venetian blind**	1 bit	`CRWTU`	[1.008] up/down	17, (1 / 1+2) 34, (2) 55, (3 / 3+4) 72, (4) 93, (5 / 5+6) 110 (6)
	Channels x and y = coupled **Type = shutter or venetian blind**				
Input *x* – **Scene number**	Channels x and y = independent, **Type = scene**	1 Byte	`CR-T-`	[17.*] Scene number [18.*] Scene control	18, (1) 35, (2) 61, (3) 73, (4) 94, (5) 111 (6)
Input *x* – **Counter value**	Channels x and y = independent, **Type = counter**	1 Byte 2 Bytes 4 Bytes	`CR-T-`	[12.001] Counter pulses [13.001] Counter pulses	18, (1) 35, (2) 61, (3) 73, (4) 94, (5) 111 (6)
Input *x* – **Counter reset command**	Channels x and y = independent, **Type = counter**	1 bit	`C-W--`	[1.015] reset	19, (1) 36, (2) 62, (3) 74, (4) 95, (5) 112 (6)
Input *x* – **Counter runout**	Channels x and y = independent, **Type = counter**	1 bit	`C-W--`	[1.055] alarm	20, (1) 37, (2) 63, (3) 75, (4) 96, (5) 113 (6)

Table 1. Independent/single channel object sizes and DPTs:

Size	*DPT*
1 bit	[1.001] switch
2 bits	[2.*] 1-bit controlled
1 byte unsigned	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte percentage	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte signed	[6.*] 8-bit signed value
2 bytes unsigned	[7.*] 2-byte unsigned value
2 bytes signed	[8.*] 2-byte signed value
2 bytes floating	[9.*] 2-byte float value

7.2 Warning

- Installation, electrical connection, configuration and commissioning of the device can only be carried out by qualified personnel
- Opening the housing of the device causes the immediate end of the warranty period
- ekinex® KNX defective devices must be returned to the manufacturer at the following address: SBS S.p.A. Via Circonvallazione s / n, I-28010 Miasino (NO) Italy

7.3 Other information

- This application manual is aimed at installers, system integrators and planners
- For further information on the product, please contact the ekinex® technical support at the e-mail address: support@ekinex.com or visit the website www.ekinex.com
- ekinex® is a registered trademark of SBS S.p.A.
- KNX® and ETS® are registered trademarks of KNX Association cvba, Brussels