merten

Binary input REG-K/4x230

Art.-No.

644992
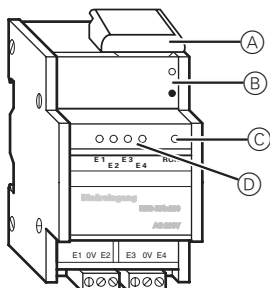
Inhaltsverzeichnis

1. Function

The Merten binary input REG-K/4x230 is used to connect conventional 230 V devices (such as movement detectors and light-sensitive switches) to the KNX bus..

i	Caution: High voltages can cause damage

2. Operating and display elements

Ⓐ   Cover of the bus connecting terminal

Ⓑ   Programming button/programming display

Ⓒ   Operating display

Ⓓ   Channel status displays

3. Installation

⚠	Risk of fatal injury: All work carried out on the unit may only be performed by skilled electricians. Observe the regulations valid in the country of use, as well as the valid KNX guidelines.
⚠	Caution: All the devices that are installed next to the binary input must be equipped with basic insulation at the very least.
⚡	Risk of fatal injury from electrical current. During installation, the safety regulations specified in DIN VDE 0105 must be observed. The device may only be installed by skilled electricians. Otherwise, there is a risk of fire or electrocution
⚠	Caution: If there are any 230 V cables in the vicinity, make sure that the distances to them specified in the applicable standards and regulations are maintained. The cover of the bus connecting terminal must always be mounted.

① Snap onto a 35 x 7.5 mm DIN profile rail which conforms to standard DIN EN 50022.
② Connect the bus plug and attach the cover of the bus connecting terminal.
③ Connect the input cables (see figure connection example)
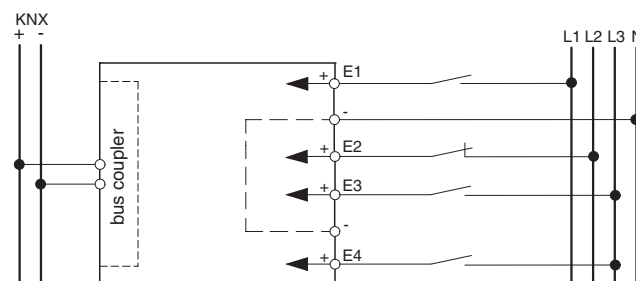
i	The N conductors must be connected to the device. Inputs E1 to E4 have a common potential (4 x N). Inputs E1 to E4 can be assigned to each other with different phases.
⚠	Caution: The connection of the potentials inside the device for binary input REG-K/8x230 is not suitable for transferring load currents.

merten

4. Technical Data

Power supply form the bus	DC 24 V/< 18 mA
Insulation voltage:	AC 4 kV bus/inputs

Inputs:

Nominal voltage::	AC 230 V ±10 %, 50/60 Hz
0 signal:	< 40 V
1 signal:	> 160 V
Nominal current:	AC approx. 7 mA

Permissible cable length:	max. 100 m/channel

Ambient temperature:

Operation:	-5 °C bis +45 °C
Storage:	-25 °C bis +55 °C
Transport:	-25 °C bis +70 °C
Max. humidity:	93 % relative humidity, no moisture condensation
Environment::	The device is designed for use at an installation height of up to 2000 m above sea level (MSL)

Connections:

Inputs, outputs:	Screw terminals
Single-core:	1.5 mm² to 2.5 mm²
Finely stranded (with connector sleeve):	1.5 mm² to 2.5 mm²
KNX	bus connecting terminal

Dimensions:

height x width x depth::	90 x 45 x 65 mm
Device width:	2.5 modules

5. Settings in the EIB Tool Software (ETS)

Selection in the product database

Manufacturer:	Merten	
Product family:	2.6	Binary Input, 4-gang
Product type:	2.6.02	DIN rail mounted REG-K/ 230 V
Range name:	Multifunction.Counter 12BA/1.0	
Media type:	Twisted Pair	
Product name:	Binary input REG-K/4x230	
Order number:	644992	

The application is ETS 3 compatible.

ℹ️ To guarantee the full functionality of the applications under ETS2, the ETS2 program from version 1.2 onwards and Service Release A or higher should be used. If you have any queries, contact the Merten infoline.

6. Application overview

Application	Vers.	Function
Multifunction.Counter 12BA/1.0	1	Switching
		Dimming or blind control via 1 or 2 inputs
		Positioning values for blind control (8-bit)
		Pulse edges with 1-, 2-, 4-, or 8-bit telegrams
		Differentiation between short/long operation
		Initialisation telegram
		Cyclical transmission
		Pulse edges with 2 bytes
		8 bit linear regulator
		Disable function
		Contact type: Make/brake contact
		Debounce time

merten

Multi-function.counter 12BA/1.0

6.0.1 Multi-function.counter 12BA/1.0

General information

You can use this application to program the binary input.

Up to two objects are available for each input.

Group addresses are managed dynamically. Maximum no. of group addresses and associations: 150.

Application functions

This application offers a wide range of setting options in order to execute numerous functions with a binary input and controlled EIB devices (e.g. dimming actuators, switch actuators etc).

Naturally, which function is possible in each individual case depends on the EIB devices being controlled. The functions of this application described here can therefore only apply to those specific EIB control functions. Here, only those tabs and parameters which are of relevance to these control functions are described.

You will find an overview of all the tabs, parameters and the related adjustable values in the last section "Parameters and settings".

i The following description assumes that push-buttons are connected to the inputs of the binary input.

i Adjustable times (staircase timer, ON delay, OFF delay etc.) are adjusted via the time base and time factor parameters. The actual time is calculated by multiplying both values; e.g. time base 1 second times time factor 3 gives 3 seconds.
If only one of these parameters is shown, no time adjustment is possible for the parameter setting selected.

Basic settings

Operating mode

For the input functions, you can select between the "Make contact" and "Break contact" operating modes.

Tab	Parameter
Input X	**Operating mode**

Additional startup delay of the application

You can start up the application later after the bus voltage has been switched on.

Tab	Parameter
General	**Additional startup delay of the application**

Debounce time

You can set a debounce time.

Tab	Parameter
General	**Debounce time**

Transmit 1/8 bit toggle commands

You can address two actor groups with 1 or 8 bits (1 byte) simultaneously.

With a 1 bit object type, the object value is first inverted with each push-button action, then sent on the bus, i. e. a "0" becomes a "1", and when the same key is pushed again, a "1" becomes a "0". The device is therefore switched on and off alternately. This switching behaviour is called "toggling". An update or change to the 1-bit/1-byte object value is possible via the bus when another sensor switches the actuator (e .g. via a two-way circuit or a central command). To prevent "incorrect" toggling, you must load the status of the actuator ("1" or "0") into the push-button interface. To do this, connect the group address of the second sensor to the switch/value object of the push-button interface.

For 1-byte object types, you can set two values, which are transmitted alternately after each push-button action.

Two objects can also be sent in any combination when the push-button is activated (1 bit / 1 byte).

Tab	Parameter
Input X	Functional selection
	Number of objects
	Object A/B
	Value 1/2

merten

Communication objects

You can select the following communication objects:

Per input:

Function	Object name	Type	Prio	Flags	Behaviour
Input X	Switch object A/B	1 bit	Low	WCT	Transmit/re-ceive

Transmit 1/8 bit switching commands

You can address two actuator groups with 1 or 8 bits simultaneously.

Depending on the parameter settings, one of the following will be transmitted via the switch/value object whenever a key is pressed:

– an ON or OFF telegram
– 1 byte values (0 % - 100 % in levels)
– 1 byte values (0 - 255) infinitely
– two objects, (1 bit / 1 byte) in any combination

sent via the switch/value object.

Tab	Parameter
Input X	Functional selection
	Number of objects
	Object A/B
	Value

Communication objects

You can select the following communication objects:

Per input:

Function	Object name	Type	Prio	Flags	Behaviour
Input X	Switch object A/B	1 bit	Low	WCT	Transmit/re-ceive

Dimming

You can use the dimming function for the following:
– dim brighter **and** darker via **one key** (single-surface dimming)
– either dim brighter **or** darker. You need a second key (second input) to dim in the other direction (dual-sur-face dimming).

You can use the corresponding key (input) to switch the light on or off (press key briefly) or dim it (press key for a longer period, the parameters for the exact period can be set). When switching takes place, an ON/OFF telegram is sent via the switch object. When dimming, dimming up or dimming down is carried out via the 4-bit dimming object; the parameters for the dimming steps can be set. In addition, you can also transmit the corresponding dimming step cyclically for a period of time which can be set as required.

Common parameters for single-surface and dual-surface dimming

Tab	Parameter
Input X	Functional selection
	Detection of prolonged activati-on 100 ms * Factor (4-250)
	Dimming direction
	Cyclical sending of the dimming levels
	only with cyclical transmission of the dimming steps: Base for cyclic interval
	only with cyclical transmission of the dimming steps: Factor for cyclic interval (3-255)

Additional parameters for single-surface dimming

You can dim brighter or darker and also switch on or off using a single key.

The current switching or dimming direction is always dependent on the previous action, i. e. if switched off, pressing the key briefly will switch the light on and vice versa, and if the light has been dimmed up, prolonged activation of the key will dim the light down again. On release after prolonged activation, a stop telegram will be sent via the 4-bit dimming object, thus terminating the dimming procedure in the dimming actuator.

An update or change to the switch/object value is possible via the bus when another sensor switches or dims the actuator (e.g. via a two-way circuit or a central command). To prevent the "wrong" switching/dimming activity, you must load the status of the actuator into the push-button interface. To do this, connect the group address of the second sensor to the switch/dimming object of the push-button interface.

A single command is sufficient to cycle through the dimming range. This dimming procedure can be used for most applications. The other possible dimming steps (1/2 - 1/64 brighter or darker) dim brighter or darker by the selected step. For example, to dim from min. to max. brightness, you would need to push the key for a prolonged period four times in succession if the level set is 1/4.

Tab	Parameter
Input X	Dimming direction
	Dimming steps (brighter)
	Dimming steps (darker)

"Dimming direction" parameter value for single-surface dimming:
– brighter and darker

merten

Additional parameters for dual-surface dimming

These are used to dim either brighter or darker and to either switch on or off using a single key. Therefore, you must set the parameters for a second key (second input) for the opposite direction.

You can set whether a stop telegram is to be transmitted when the key is released. When you have enabled the transmission of a stop telegram, a stop telegram will be sent via the 4-bit dimming object after prolonged activation of the key, thus terminating the dimming procedure in the dimming actuator.

A single command is sufficient to cycle through the dimming range. This dimming procedure can be used for most applications. The other possible dimming steps (1/2 - 1/64 brighter or darker) dim brighter or darker by the selected step. For example, to dim from min. to max. brightness, you would need to push the key for a prolonged period four times in succession if the level set is 1/4.

Tab	Parameter
Input X	Dimming direction
	only in the dimming direction "brighter": dimming steps (brighter)
	only in the dimming direction "darker": dimming steps (darker)
	Stop telegram after release

"Dimming direction" parameter value for dual-surface dimming:
– brighter
– darker

Communication objects

You can select the following communication objects:

Per input:

Function	Object name	Type	Prio	Flags	Behaviour
Input X	Switch object	1 bit	Low	WCT	Transmit/receive
Input X	Dimming object	4 bit	Low	WCT	Transmit/receive

Blind control

You can use the blind control function to do the following:
– move the blind using an individual key and adjust the slats (single-surface blind operation).
– With the blind control function, you can raise the blinds / adjust the slats using a single key and lower the blinds / adjust the slats using a second key (dual-surface blind operation).
– Move the blind to a pre-specified position.
– Move the blind between two previously specified positions.

Tab	Parameter
Input X	Functional selection

Blind control function up or down with one key in each case (dual-surface blind operation)

After the corresponding key is pressed for a short time, a stop/step telegram will be transmitted; after the key is activated for a prolonged period (the parameters for the exact period can be set), a movement telegram will be transmitted. With this function, you must set the parameters for a second key (second input) with the corresponding settings for the blind movement in the opposite direction. Both keys (inputs) must be given the same group addresses.

Tab	Parameter
Input X	Functional selection
	Detection of prolonged activation from 100 ms * Factor (4-250)
	Direction of movement

Blind control function up or down with a single key (single-surface blind operation)

The current direction of movement of the blind, or the direction of the slat adjustment, always depends on the previous action, i. e. when the blind has just been moved downwards, it will move upwards the next time the key is activated for a long period (parameters for the period can be set).

After a stop/step telegram has been transmitted to adjust the slats, a stop/step telegram for the same direction of movement can be created by pressing the key again, as long as this subsequent push-button action is carried out within a time period, the parameters for which can be set. If this time period has elapsed, the direction of rotation of the slats will change when the key is pressed briefly.

merten

Multi-function.counter 12BA/1.0

The push-button (input) can receive telegrams via the stop/step movement objects, and can create corresponding telegrams when the key is pressed, depending on the values received. An update or change to the switch/object value is possible via the bus when another sensor switches the actuator (e .g. via a two-way circuit or a central command). To prevent "incorrect" movement, you must load the status of the actuator into the push-button. To do this, connect the group address of the second sensor to the stop/step and the movement object of the push-button interface.

Tab	Parameter
Input X	Detection of prolonged activation from 100 ms * Factor (4-250)
	Direction of movement
	Change in direction for slat adjustment from 100 ms * Factor (5-50)

Move the blind to a pre-specified position.

If the blind actuator is capable of approaching a specific position, you can define one or two positions using this function, which can be approached by the blind using 1-byte position values with a push-button action. The position values can be set in steps between 0% and 100%, or infinitely from 0-255.

When approaching a position, the set value for the blind position and the slat position is transmitted using a short (or long) push-button action.

To trigger two positions, enter the required blind position and slat position for both. Position value 1 is transmitted with a short push-button action, while position value 2 is transmitted with a long push-button action. No movement or stop/step objects exist with these set parameters.

Tab	Parameter
Input X	Direction of movement
	Selecting the positioning
	Position value 1 (press briefly)
	Value for blind position
	Value for slat position
	only with "two positions": Position value 2 (press for long period)

Communication objects

You can select the following communication objects:

Per input:

Function	Object name	Type	Prio	Flags	Behaviour
Input X	Stop/step object	1 bit	Low	CT	Transmit/receive
Input X	Movement object	1 bit	Low	CT	Transmit/receive
Input X	Blind position	1 byte	Low	CT	Transmit/receive
Input X	Slat position	1 byte	Low	CT	Transmit/receive

Transmit 1 bit, 2 bit (priority control), 4 bit or 1 byte pulse edge commands

You can use these pulse edge functions to parameterise different object actions. You can transmit one or two objects simultaneously, and select the size of the objects required (1 bit, 2 bit priority control, 4 bit or 1 byte in steps or infinitely) as needed. This enables you to parameterise a large number of application options.

You can specify which actions should be carried out when a key is pressed, and which should be carried out when a key is released. These actions could include:

– Transmit 1 or 0 (with 1 bit)

– Transmit value 1 or value 2 (with 2 bits, 4 bits or 1 byte):
  You can enter two values and set whether and how they are to be transmitted.

– Transmitting a value:
  The object transmits the value which it has currently been given. You can therefore transfer a value e. g. with the transmitting group address which was previously received by another group address.

– Toggling:
  The current object value is inverted and then transmitted. It is therefore switched on and off alternately (toggled). The value can be modified via the bus.

– no action

Tab	Parameter
Input X	Functional selection
	Pulse edge function
	only with extended pulse edge function: Detection of prolonged activation from 100 ms * Factor (4-250)
	Number of objects

merten

Chapter 4: Binary inputs

4.2 Binary inputs, 4-gang
Binary input REG-K/4x230
Art. no. 644992

Multi-function.counter 12BA/1.0

Normal pulse edge function

With the standard pulse edge function, you can transmit 1 bit, 2 bit, 4 bit priority control or 1 byte in steps or infinitely.

| Tab	Parameter
Input X - (object A&B)	Object A/B
	Action on activation
	Action on release
	only with 2 bit and 4 bit (priority control): value 1 / value 2
	only with 1 byte in steps 0 - 100 % value 1 / value 2
	only with 1 byte infinitely 0-255: value 1 / value 2

Principle of the pulse edge function

Using the following diagrams, you can see how the pulse edge function behaves when pulse edges rise or fall.
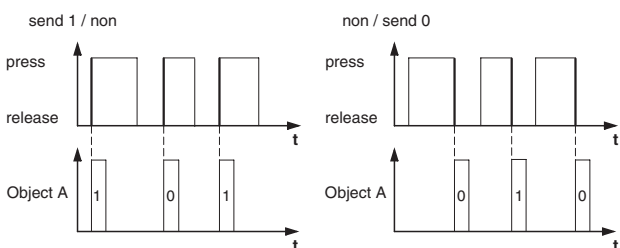
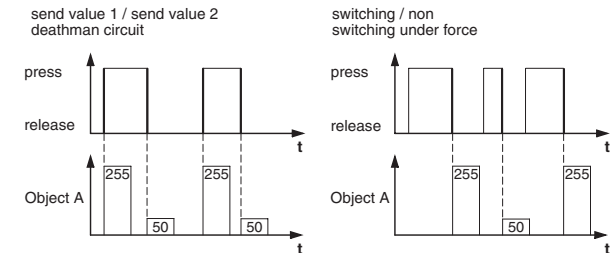Object A = 1 Bit
action at pressed/released
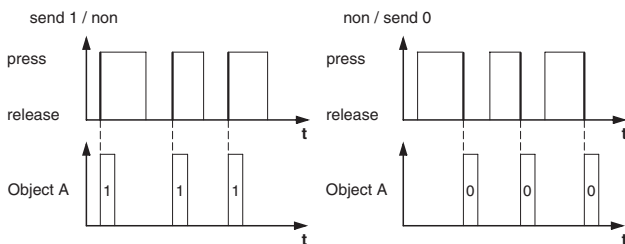
Object A = 1 Bit
action at pressed/released
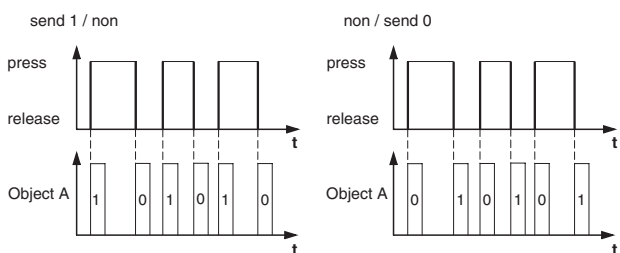
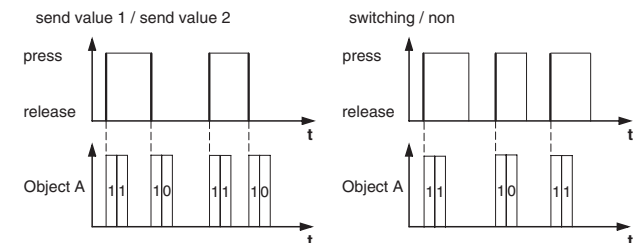
Object A = 1 Bit
action at pressed/released

Object A = 1 Byte variable 0-255
value 1 = 255
value 2 = 50
action at pressed/released

Example: Function "death man circuit" or "switching under force"

Object A = 2 Bit (guidance under force)
value 1 = 11 (switching on under force)
value 2 = 10 (switching off under force)
Action at pressed/released

Extended pulse edge function

With the extended pulse edge function, you have a wider range of functions available, e. g. you can set different actions with a shorter or longer activation time, which apply to both the actions which result when the key is pressed and when the key is released. You can also set a cycle time which can be parameterised for each object.

> When parameterising, bear in mind that you need to set all four types of key activation (brief/long, pressing and releasing the key) in order to ensure that the push-button (input) functions as required.

Tab	Parameter
Input X - (object A&B)	Object A/B

merten

Multi-function.counter 12BA/1.0

A description of the most important actions is given below:

– transmits [value]:
transmits the current value and stops a cyclical transmission.

– transmits [value] immediately and then cyclically:
If no cycle time is running, [value] is transmitted immediately and a new cycle time is started. If a cycle time is already running, this is interrupted, [value] is transmitted and a new cycle time is started.

– transmits [value] only cyclically:
If no cycle time is running, [value] is transmitted immediately and a new cycle time is started. If a cycle time is already running, this is **not** interrupted, [value] is transmitted after the current cycle time has elapsed, and a new cycle time is started.

– sets object value to [value] (readable only)
[value] is written into the object and is not transmitted An active cycle time is terminated.

– toggles:
compares the current object value with [value]. If both are the same, value 1 or value 2 is transmitted. If both are different, [value] is transmitted.

– toggles, transmits immediately, then cyclically:
The value is toggled (see "toggles") if no cycle time is running, transmitted immediately and a new cycle time is started. If a cycle time is already running, this is interrupted, the toggled value is transmitted and a new cycle time is started. Subsequently, the value which has already been toggled is always transmitted cyclically.

– toggles, only transmits cyclically:
If no cycle time is running, the toggled value is transmitted immediately and a new cycle time is started. If a cycle time is already running, this is **not** interrupted, the toggled value is transmitted after the current cycle time has elapsed, and a new cycle time is started. Subsequently, the value which has already been toggled is always transmitted cyclically.

– toggles and is not transmitted:
The toggled value is written into the object and is not transmitted. An active cycle time is terminated.

– toggles cyclically, transmits immediately, then cyclically:
The value is toggled (see "toggles") if no cycle time is running, transmitted immediately and a new cycle time is started. If a cycle time is already running, this is interrupted, the toggled value is transmitted and a new cycle time is started. It is always subsequently cyclically toggled, and the new value is transmitted.

– toggles cyclically, only transmits cyclically:
If no cycle time is running, the toggled value is transmitted immediately and a new cycle time is started. If a cycle time is already running, this is **not** interrupted, the toggled value is transmitted after the current cycle time has elapsed, and a new cycle time is started. It is always subsequently cyclically toggled, and the new value is transmitted.

– toggles cyclically and is not transmitted:
The toggled value is written into the object and is **not** transmitted. It is always subsequently cyclically toggled, and the new value is written into the object.

– transmits its value:
The current object value is transmitted. An active cycle time is terminated.

– sends its value immediately and then cyclically:
If no cycle time is running, the current object value is transmitted immediately and a new cycle time is started. If a cycle time is already running, this is interrupted, the current object value is transmitted and a new cycle time is started. Subsequently, the current object value is always transmitted cyclically.

– increase the current object value by [value] cyclically:
If no cycle time is running, [value] is added to the current object value, the object value is transmitted, and a new cycle time is started. If a cycle time is already running, this is **not** interrupted, the current object value with [value] added is transmitted and a new cycle time is started.

– reduce the current object value by [value] cyclically:
If no cycle time is running, [value] is subtracted from the current object value, the object value is transmitted, and a new cycle time is started. If a cycle time is already running, this is **not** interrupted, the current object value with [value] subtracted is transmitted and a new cycle time is started.

– transmits [valueA] and after a cycle time [valueB]:
[ValueA] is transmitted immediately, and [valueB] is transmitted after **one** cycle time, regardless of whether a cycle time is already running or not (staircase timer function).

– none (stops cyclical transmission):
No action is carried out, and any active cycle time is stopped.

– no change:
The current action remains unchanged (e. g. "transmits value1 and after a cycle time, transmits value2").

– none (stop after current cycle time has elapsed):
No action is currently carried out, but any active cycle time is **not** stopped. It runs through until the end, and then transmits the corresponding value.

Examples of use for the pulse edge function

The following activation sequence diagram shows the phases into which the pulse edge function is divided:
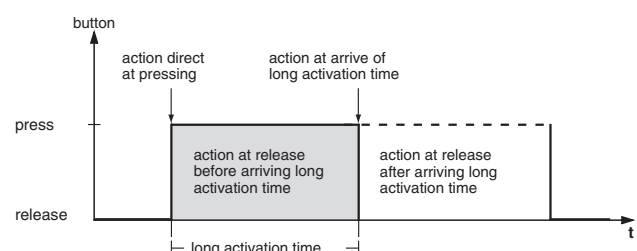
Example: Staircase lighting function with cleaning lighting function

With a brief push-button action, the switch actuator switches on the light. A long push-button action extends the staircase lighting function (= cleaning lighting function) until a second, long push-button action switches off the actuator. The switch actuator requires a staircase lighting function and a disable function for this function.
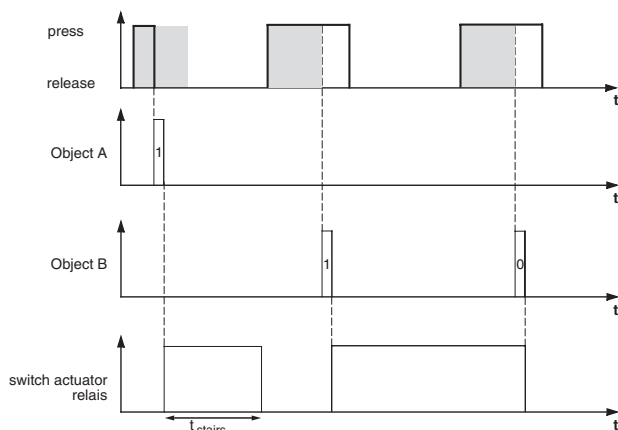
Number of objects = 2 (object A/B)

Object A/B = 1 bit

Object A: Action on release before the long activation time has elapsed = transmits 1

Object B: Action on completion of the long activation time = toggles

To do this, connect object A with the switch object and object B with the disable object of the switch actuator.

Example: short and long staircase time

You can use this function to produce a brief and a long staircase time with the push-button. The switch actuator requires no staircase lighting function for this request.

With a brief push-button action, the switch actuator switches on the light, and after a parameterised cycle time (e. g. 3 minutes), it switches it back off again. With a long push-button action, the same function is carried out, but with a longer cycle time (e. g. 6 minutes).

Number of objects = 2 (object A/B)

Object A/B = 1 bit

Object A: Action on release before the long activation time has elapsed = transmits 1. After a cycle time has elapsed (here 3 minutes) = transmits 0

Object B: Action on release when the long activation time is completed = transmits 1. After a cycle time has elapsed (here 6 minutes) = transmits 0

To do this, connect object A and object B with the switch object of the switch actuator.

Example: Switch the light on/off permanently, or switch off after a cycle time has elapsed.

With a brief push-button action, the switch actuator switches the light on or off permanently. With a long push-button action, the light switches on, and after a parameterised cycle time (e .g. 6 minutes), it switches back off again. Due to the cycle time in the push-button which can be parameterised, the switch actuator requires no staircase lighting function for this function.

Number of objects = 2 (object A/B)

Object A/B = 1 bit

Object A: Action on release before the long activation time has elapsed = toggles

Object B: Action when the long activation time is completed = transmits 1. After a cycle time has elapsed (here 6 minutes) = transmits 0. Action on release when the long activation time is completed = no change.

To do this, connect object A and object B with the switch object of the switch actuator.

merten

Example: electronic protection against theft

This example will show you how to program electronic protection against theft for the push-button. It is activated by a brief push-button action and then transmits cyclically. As soon as the push-button is forcibly separated from the push-button interface, this can be shown on a display, or an alarm can be triggered.
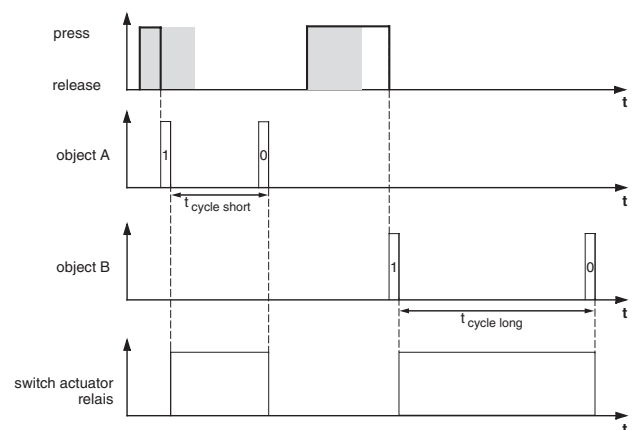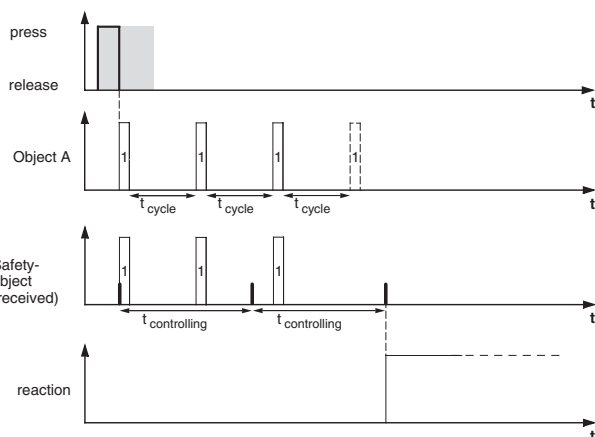
Number of objects = 1 (object A)

Object A = 1 bit

Object A: Action on release before the long activation time has elapsed = transmits 1 immediately and then cyclically. Action when the long activation time is completed = no change. Action on release after the long activation time is completed = no change. Cycle time = e. g. 10 minutes

To do this, connect object A with an object which anticipates cyclical telegrams (e. g. a safety object). The monitoring time set on the safety object must be longer than the cycle time of the push-button. If the safety object receives no telegrams from the push-button during this time, a reaction which can be parameterised is activated (e. g. the channel is switched on).

Example: Effect lighting

This example shows you how to program effect lighting, for example for a display window. A long push-button action switches between two different lighting scenes. A short push-button action stops the toggling and transmits a scene (to retrieve the scene, use the scene module for the actuator which has been activated) which switches off everything.

Number of objects = 2 (object A/B)

Object A/B = 1 byte infinite 0-255

Object A: Direct action when activated = none (stops cyclical transmission). Action on release before the long activation time has elapsed = transmits 1. Action when the long activation time is completed = none (stops cyclical transmission). Action on release after the long activation time is completed = none (stops cyclical transmission). Value 1 = 3.

Object B: Direct action when activated = none (stops cyclical transmission). Action on release before the long activation time has elapsed = none (stops cyclical transmission). Action when the long activation time is

completed = none (stops cyclical transmission). Action on release after the long activation time has been completed = toggles cyclically, transmits immediately, then cyclically. Value 1 = 1, value 2 = 2nd cycle time = e. g. 1 minute.

To do this, connect object A and object B with the extension unit object of the scene function.
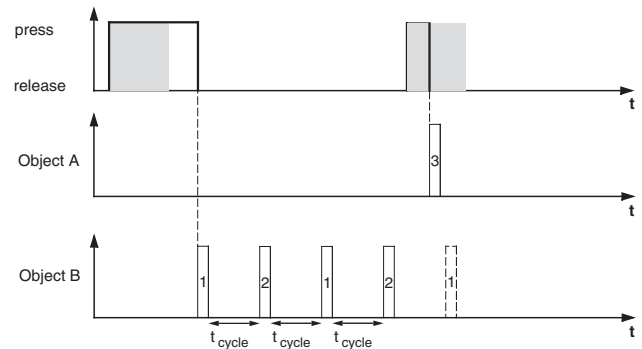
Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Input X	Object A/B	1 bit	Low	WCT	Transmit/receive
Input X	Object A/B	2 bit	Low	WCT	Transmit/receive
Input X	Value object A/B	1 byte	Low	WCT	Transmit/receive

merten

Multi-function.counter 12BA/1.0

Transmit 2 byte pulse edge commands via an object

You can also use these pulse edge functions to parameterise different object actions. However, in contrast to pulse edge functions, you can only transmit one object with 1 bit, 2 bits, 4 bits or 1 byte.

You can transmit normal or extended pulse edge commands. With normal pulse edge commands, you can specify which actions should be carried out when a key is pressed, and which should be carried out when a key is released. With extended pulse edge commands, you can also set the actions before and after the long activation time is completed.

You can transmit floating point numbers and whole numbers, without or without a sign.

Tab	Parameter
Input X	Functional selection
	Pulse edge function
	only with extended pulse edge function: Detection of a long activation time from 100 ms * Factor (4-250)
	Action when activated
	Action on release *only with extended pulse edge function:*
	Action on release before the long activation time has elapsed
	Action when the long activation time is completed
	Action on release after the long activation time has been completed

Explanation of actions: See 1-bit pulse edges.

Tab	Parameter
Input X - values	Object type value
	Only with floating point: value 1/2 Basic value, adjustable value range in brackets
	only with whole number with sign: value 1/2 (-32768 - 32767)
	Only with floating point: value 1/2 (0 - 65535)

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Input X	Value object A	2 byte	Low	WCT	Transmit/receive

Setting the parameters for the 8-bit linear regulator

You can use the 8-bit linear regulator function to program a key (input) as a linear regulator. You can parameterise all four actions when pressing/releasing with a shorter or longer operating time in each case. You can establish the function with or without limit values (start/end value).

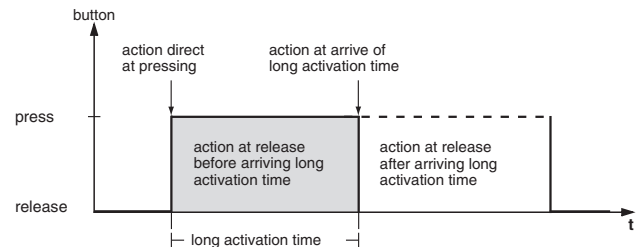| Tab	Parameter
Input X	Functional selection
	Detection of long activation time from 100 ms * Factor (4-250)
Input X (2	Linear regulator function
	Direct action when operated
	Action on release before the long activation time has elapsed
	Action when the long activation time is completed
	Action on release after the long activation time has been completed
	only with "start value and end value" Start value
	Step value
	only with "start value and end value" end value
	Base for cyclic interval
	Factor for cyclic interval (3-255)

merten

A description of the actions is given below:

– Transmit the start value, then increase cyclically by the step value:
  If no cycle time is running, the start value is transmitted immediately and a new cycle time is started. If a cycle time is already running, this is interrupted, the start value is transmitted and a new cycle time is started.

– Transmit the end value, then reduce cyclically by the step value:
  If no cycle time is running, the end value is transmitted immediately and a new cycle time is started. If a cycle time is already running, this is interrupted, the end value is transmitted and a new cycle time is started.

– Increase the current object value cyclically:
  Increase the current object value cyclically by the parameterised step interval.

– Increase the current object value once:
  Increase the current object value once by the parameterised step interval. An active cycle time is terminated.

– Reduce the current object value cyclically:
  Reduce the current object value cyclically by the parameterised step interval.

– Reduce the current object value once:
  Reduce the current object value once by the parameterised step interval. An active cycle time is terminated.

– Reverse the slide direction and transmit cyclically:
  If no cycle time is running, the slide is pushed in the opposite direction (of this push-button) and a new cycle time is started. If a cycle time is already running, it is interrupted, the slide is immediately pushed in the opposite direction (of this push-button) and a new cycle time is started.

– Move step-by-step to the limit values, and then back again:
  The limit values are approached by one step interval at a time. When a limit is reached, the slide direction is reversed for the next action.

– none (stops cyclical transmission):
  No action is carried out, and any active cycle time is stopped.

– no change:
  No action is carried out, and any active cycle time is continued.

> **i** You can only maintain the limit values and the toggling to a new slide direction by operating on site!

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Input X	Value object A	1 byte	Low	WCT	Transmit/receive
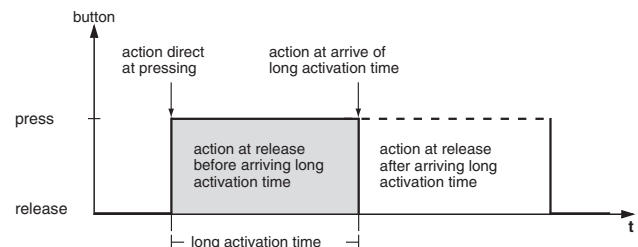
Retrieving scenes

The scene retrieval function does not access the internal scene module, but only accesses the bus externally via communication objects.

There are two types of scene function:

– normal
– extended

With the standard scene function, a scene is retrieved by a brief push-button action while a long push-button action is used to save a scene. You merely have to set the time after which a push-button action is identified as being long, together with the status LED control and the scene address.

Tab	Parameter
Input X	Functional selection
	Detection of long activation time from 100 ms * Factor (4-250)
	Scene function
	Only with "extended" Number of objects
	Only with "normal scene function" Scene address (0-63)

With the extended scene function, a wider range of functions are available. You can set actions for a shorter or longer activation time for both pressing and releasing the keys. You can also program a cycle time.

Depending on how many objects you have set in the scene function, you must make the settings in additional windows ("Input X - object A") or ("Input X - object B"):

merten

Multi-function.counter 12BA/1.0

Tab	Parameter
Input X - (object A/B)	Direct action when activated
	Action on release before the long activation time has elapsed
	Action when the long activation time is completed
	Action on release after the long activation time has been completed
	Value 1 Scene address (0-63)
	Scene addresses are designed for the scene
	Value 2 Scene address (0-63)
	Scene addresses are designed for the scene
	Base for cyclic interval
	Factor for cyclic interval (3-255)

Communication objects

You can select the following communication objects:
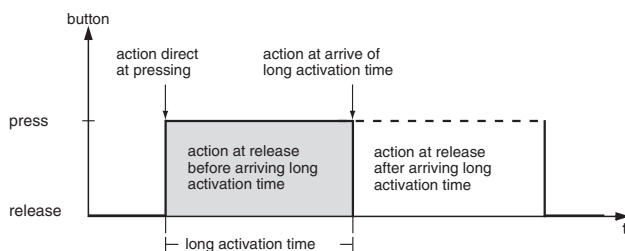
Function	Object name	Type	Prio	Flags	Behaviour
Input X	Object A/B	1 byte	Low	WCT	Transmit/receive

Activate impulse counter

You can use the impulse counter function to cyclically count the activation of the input within a parameterisable time frame and transmit this as a value (2-byte value object) via the bus. A count can be made up to 65535 (16 bits).

After the time frame has elapsed, the current counter reading is transmitted, the counter is reset and the counting process begins again.

The impulse counter can also be reset via another input which is programmed accordingly as a synchronous input.

The current counter reading can be read by setting the read flag.

i Ensure that the counter reading is not overwritten via the bus (write flag)!

You can determine whether the count should be made while the pulse edge is rising or falling, or while it is both rising and falling.

You can parameterise the cyclic interval using the cycle time base and a cycle time factor (cycle time = cycle time base x cycle time factor).

Tab	Parameter
Input X	Functional selection
	Count the impulses when
	Base for cyclic interval
	Factor for cyclic interval (3-255)

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Impulse counter	Value object A	2 byte	Low	RWCT	Transmit

Activating the switching counter

You can use the switching counter function to cyclically count the activation of the input and after a parameterisable counter reading, transmit this as a value (2-byte value object) via the bus. In addition, you can transmit a switching telegram when the counter reading is reached.
A count can be made up to 65535 (16 bits).

After the set counter reading is reached, the current counter reading is transmitted and the counter reading is reset to 0.

The current counter reading can be read off by setting the read flag.

i Ensure that the counter reading is not overwritten via the bus (write flag)!

Additionally, you have the option of transmitting the counter reading cyclically after an adjustable number of counting impulses.

You can determine whether the count should be made while the pulse edge is rising or falling, or while it is both rising and falling.

Tab	Parameter
Input X	Functional selection
	Count the impulses when
	Command when maximum counter reading is reached
	Maximum counter reading (1-65535)
	Gradual counter output (1-255)

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Switching counter	Value object A	2 byte	Low	WCT	Transmit
Switching counter	Switch object A	1 bit	Low	WCT	Transmit

merten

Multi-function.counter 12BA/1.0

Resetting the counter

You can use the counter reset function to reset the impulse counter and the cycle time or the switching counter of another input. Additionally, you can use this function to transmit switching commands.

You can determine whether the count should be reset while the pulse edge is rising or falling, or while it is both rising and falling.

Additionally, you can program a switching telegram to be transmitted when the pulse edge is rising and/or falling.

Tab	Parameter
Input X	Functional selection
	Influences input (1-4)
	Behaviour on rising pulse edge
	Behaviour on falling pulse edge
	Counter and cyclic interval

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Reset counter	Switch object A	1 bit	Low	WCT	Transmit

Cyclical transmission for devices with cyclical monitoring

You can use the cyclical monitoring function to transmit telegrams cyclically. This is necessary when other devices are monitoring the presence of the push-button interface via their cyclical telegrams (protection against theft, fault monitoring). If no telegram is received from the push-button interface within a parameterisable time period, e. g. a signal is issued by this device. For this purpose, you can parameterise the push-button interface accordingly.

Tab	Parameter
Input X	Functional selection
	Operating mode
	Number of objects

Tab	Parameter
Input X (object A/B)	Object A/B
	Action when activated
	Action on release
	Value 1
	Value 2
	Base for cyclic interval
	Factor for cyclic interval (3-255)

i When a disable function is activated via the disable object, all current push-button functions are reset or interrupted.

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Cyclical monitoring	Object A/B	1 bit/ 2 bit/ 4 bit/ 1 byte	Low	WCT	Transmit

Activating the disable function for the inputs

You can block the inputs in two different ways:

1. Separately for each input
2. All inputs function as a master input

You can determine whether a disable object = 0 or = 1 should be blocked.

Tab	Parameter
Input disable function	Disable function
	Blocking
	Behaviour at start of blocking
	Master input

Separately for each input

You can use this function to block each of the four or eight inputs individually. When an input is blocked, it fulfils no function. You can use an additional parameter to parameterise the behaviour of cyclical functions.

Tab	Parameter
Inputs X: Block	Input x
	Cyclical actions are

All keys function as a master input

You can use this action to specify one of the two or four inputs as the master input. When any key is pressed, the action which has been parameterised for the master key is carried out.

Tab	Parameter
Input X	Functional selection
	Type of blocking
	Master input

merten

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Disable function	Disable object	1 bit	Low	WC	Receive

i When a disable function is activated via the disable object, all current push-button functions are reset or interrupted.

Behaviour when bus voltage is applied/restored or fails

Behaviour on application/recovery of the bus voltage

When a bus voltage is applied or recovered, telegrams can be sent depending on the setting.

merten

Parameters and settings

General	
Parameter	**Setting**
Additional startup delay of the application 1s * factor	0 - 30, **0** default setting
Debounce time 10ms * factor	2 - 15, **2** default setting

Input X	
Parameter	**Setting**
Functional selection	Toggle
	Switching
	Dimming
	Blind control
	Pulse edges 1 bit, 2 bit (priority), 1-byte values
	Pulse edges with 2-byte values
	8-bit linear regulator
	Scene
	Impulse counter
	Switching counter
	Reset counter
	Cyclical monitoring
Operating mode	make contact
	break contact

When "Functional selection" has been set to "Toggle" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Toggle
Number of objects	**one**
	two
Object A/B	**1 bit**
	1 byte in steps 0 % - 100 %
	1 byte infinitely 0 - 255
Value	**100 %**
	adjustable in steps of ten as well as 25% and 75%
Value	0 - 255, **255** default setting

When "Functional selection" has been set to "Switching" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Switching
Number of objects	**one**
	two
Object A/B	**1 bit**
	1 byte in steps 0 % - 100 %
	1 byte infinitely 0 - 255
Value	**ON telegram**
	OFF telegram
Value	**100 %**
	adjustable in steps of ten as well as 25% and 75%
Value	0 - 255, **255** default setting

When "Functional selection" has been set to "Dimming" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Dimming
Detection of long activation time 100 ms * Factor (4-250)	4 - 250, **6** default setting
Dimming direction	brighter
	darker
	brighter and darker
Dimming steps (brighter)	**to max. brightness**
	1/2 brighter
	1/4 brighter
	1/8 brighter
	1/16 brighter
	1/32 brighter
	1/64 brighter
Dimming steps (darker)	**to min. brightness**
	1/2 darker
	1/4 darker
	1/8 darker
	1/16 darker
	1/32 darker
	1/64 darker
Cyclical transmission of the dimming steps	yes
	no
Base for cyclic interval	**0.1 seconds**
	1 second
	1 minute
	1 hour
	1 day
Factor for cyclic interval (3-255)	3 - 255, **8** default setting
Stop telegram after release	**enabled**
	disabled

When "Functional selection" has been set to "Blind control" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Blind control
Detection of long activation time from 100 ms * Factor (4-250)	4 - 250, **6** default setting
Direction of movement	UP
	DOWN
	up and down
	with position values
Change in direction for slat adjustment from 100 ms * Factor (5-50)	5 - 50, **10** default setting
Selecting the positioning	one position (press briefly)
	two positions (differentiation between short/long operation)
Position value 1 (short operation)	**in steps of 0 %- 100 %**
	infinitely 0-255
Value for blind position	*with "steps":* 0 % - 100 % in steps of 10, **100 %** default setting
	with "infinitely": 0 - 255 in single steps, **255** default setting

merten

Input X	
Parameter	**Setting**
Value for slat position	with "steps": 0 % - 100 % in steps of 10, **0 %** default setting
	with "infinitely": 0 - 255 in single steps, **0** default setting
Position value 2 (long operation)	**in steps of 0 %- 100 %**
	infinitely 0-255

When "Functional selection" has been set to "Pulse edges 1 bit, 2 bit (priority), 1-byte values" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Pulse edges 1 bit, 2 bit (priority), 1-byte values
Pulse edge function	**normal (pressed, released)**
	extended (long and short operation)
Detection of long activation time from 100 ms * Factor (4-250)	4 - 250, **6** default setting
Number of objects	**one**
	two
Object A/B	**1 bit**
	2 bit (priority control)
	1 byte in steps 0 % - 100 %
	1 byte infinitely 0 -255
Action when activated	*only with 1 bit:* **transmits 1**
	only with 1 bit: transmits 0
	only with 2 bit/1 byte: transmits value 1
	only with 2 bit/1 byte: transmits value 2
	toggles
	transmits its value
	none
Action on release	*only with 1 bit:* transmits 1
	only with 1 bit: **transmits 0**
	only with 2 bit/1 byte: transmits value 1
	only with 2 bit/1 byte: transmits value 2
	toggles
	transmits its value
	none
value 1 / value 2	switch on with priority control (11)
	switch off with priority control (10)
	switch off priority control (00)
value 1 / value 2	0 -100 % in 10% steps **100 %** default setting value 1, **0 %** default setting value 2
value 1 / value 2	0- 255, **255** default setting value 1, **0** default setting value 2

Input X - (object A/B)	
Parameter	**Setting**
Object A/B	**1 bit**
	2 bit (priority control)
	1 byte in steps 0 % - 100 %
	1 byte infinitely 0 -255

merten

Multi-function.counter 12BA/1.0

Input X - (object A/B)	
Parameter	**Setting**
Direct action when activated	*only with 1 bit:*transmits 1
Action on release before the long activation time has elapsed	*only with 1 bit:* transmits 1 immediately and then cyclically
Action on completion of the long activation time	*only with 1 bit:* transmits 1 only cyclically
	only with 1 bit: sets object value to 1 (readable only)
Action on release after completion of the long activation time	*only with 1 bit:*transmits 0
	only with 1 bit: transmits 0 immediately and then cyclically
	only with 1 bit: transmits 0 only cyclically
	only with 1 bit: sets object value to 0 (readable only)
	only with 2 bit/1 byte: transmits value 1
	only with 2 bit/1 byte: transmits value 1 immediately and then cyclically
	only with 2 bit/1 byte: transmits value 1 only cyclically
	only with 2 bit/1 byte: sets object value to value 1 (readable only)
	only with 2 bit/1 byte: transmits value 2
	only with 2 bit/1 byte: transmits value 2 immediately and then cyclically
	only with 2 bit/1 byte: transmits value 2 only cyclically
	only with 2 bit/1 byte: sets object value to value 2 (readable only)
	toggles
	toggles, transmits immediately, then cyclically
	toggles, only transmits cyclically
	toggles and is not transmitted
	toggles cyclically, transmits immediately, then cyclically
	toggles cyclically, only transmits cyclically
	toggles cyclically and is not transmitted
	only with 1 bit: transmits its value
	only with 1 bit: transmits its value immediately and then cyclically
	only with 1 bit: transmits 1 immediately and after a cycle time of 0
	only with 2 bit/1 byte transmits value 1 and after a cycle time value 2
	only with 1 byte increase the current object value by value 1 cyclically
	only with 1 byte reduce the current object value by value 2 cyclically
	none (stops cyclical transmission)
	no change
	none (stop after current cycle time has elapsed)

Input X - (object A/B)	
Parameter	**Setting**
value 1 / value 2	switch on with priority control (11)
	switch off with priority control (10)
	switch off priority control (00)
value 1 / value 2	0 -100 % in 10% steps **100 %** default setting value 1, **0 %** default setting value 2
value 1 / value 2	0- 255, **255** default setting value 1, **0** default setting value 2
Base for cyclic interval	**0.1 seconds**
	1 second
	1 minute
	1 hour
	1 day
Factor for cyclic interval (3-255)	3-255, **10** default setting

When "Functional selection" has been set to "Pulse edges with 2-byte values" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Pulse edges with 2-byte values
Pulse edge function	**normal (pressed, released)**
	extended (long and short activation)
Detection of long activation time from 100 ms * Factor (4-250)	4 - 250, **6** default setting
Action when activated	**transmits value 1**
	only with extended pulse edge function: transmits value 1 immediately and then cyclically
	only with extended pulse edge function: transmits value 1 only cyclically
	only with extended pulse edge function: sets object value to value 1 (readable only)
	transmits value 2
	only with extended pulse edge function: transmits value 2 immediately and then cyclically
	only with extended pulse edge function: transmits value 2 only cyclically
	only with extended pulse edge function: sets object value to value 2 (readable only)
	transmits its value
	only with extended pulse edge function: transmits value 1 and after a cyclic interval value 2
	only with extended pulse edge function: none (stops cyclical transmission)
	none

merten

Input X	
Parameter	**Setting**
Action on release Action on release before the long activation time has elapsed Action when the long activation time is completed Action on release after the long activation time has been completed	transmits value 1
	only with extended pulse edge function: transmits value 1 immediately and then cyclically
	only with extended pulse edge function: transmits value 1 only cyclically
	only with extended pulse edge function: sets object value to value 1 (readable only)
	transmits value 2
	only with extended pulse edge function: transmits value 2 immediately and then cyclically
	only with extended pulse edge function: transmits value 2 only cyclically
	only with extended pulse edge function: sets object value to value 2 (readable only)
	transmits its value
	only with extended pulse edge function: transmits value 1 and after a cyclic interval value 2
	only with extended pulse edge function: **none (stops cyclical transmission)**
	none

Input X - values	
Parameter	**Setting**
Object type value	Floating point
	Whole number with sign (-32768 ... 32767)
	Whole number without sign (0 ... 65535)
Value 1/2 Basic value, adjustable value range in brackets	different values between 0.01 and 327.68 in different step intervals **0.01 (0 to 20.47)** default setting
value 1/2 (-32768 - 32767)	-32768 - 32767
value 1/2 (0 - 65535)	0 - 65535

When "Functional selection" has been set to "8-bit linear regulator" in the "Input X" tab:

Input X (2	
Parameter	**Setting**
Linear regulator function	**with start value and end value**
	without start value and end value
Direct action when activated Action on release before the long activation time has elapsed Action on when the long activation time is completed Action on release after the long activation time has been completed	*only with "start value and end value"* Transmit the start value, then increase cyclically by the step interval
	only with "start value and end value" Transmit the end value, then reduce cyclically by the step interval
	Increase the current object value cyclically
	Increase the current object value once
	Reduce the current object value cyclically
	Reduce the current object value once
	Reverse the slide direction and transmit cyclically
	Move step-by-step to the limit values and then back again
	none (stops cyclical transmission)
	no change
Start value	0 - 255 in single steps
Step value	0 - 255 in single steps
End value	0 - 255 in single steps
Base for cyclic interval	**0.1 seconds**
	1 second
	1 minute
	1 hour
	1 day
Factor for cyclic interval (3-255)	3 - 255 in single steps, **5** default setting

When "Functional selection" has been set to "Scene" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Scene
Detection of long activation time from 100 ms * Factor (4-250)	4 - 250 in single steps, **30** default setting
Scene function	**normal (short = send / long = save)**
	extended
Number of objects	**one**
	two
Scene address (0-63)	0 - 63 in single steps

merten

Multi-function.counter 12BA/1.0

Input X - (object A/B)	
Parameter	**Setting**
Action when activated	transmits value 1
	transmits value 2
	toggles
	toggles cyclically, transmits immediately, then cyclically
	transmits value 1 and after a cyclic interval value 2
	none (stops cyclical transmission)
	no change
Action on release before the long activation time has elapsed	**transmits value 1**
	transmits value 2
	toggles
	toggles cyclically, transmits immediately, then cyclically
	transmits value 1 and after a cyclic interval value 2
	none (stops cyclical transmission)
	no change
Action on when the long activation time is completed	transmits value 1
	transmits value 2
	toggles
	toggles cyclically, transmits immediately, then cyclically
	transmits value 1 and after a cyclic interval value 2
	none (stops cyclical transmission)
	no change
Action on release after the long activation time has been completed	transmits value 1
	transmits value 2
	toggles
	toggles cyclically, transmits immediately, then cyclically
	transmits value 1 and after a cyclic interval value 2
	none (stops cyclical transmission)
	no change
Scene address 1 (0-63) Scene address 2 (0-63)	0 - 63 in single steps
Scene address 1 is designed to Scene address 2 is designed to	retrieve the scene (set by default to value 1)
	store scene (set by default to value 2)
Base for cyclic interval	0.1 seconds
	1 second
	1 minute
	1 hour
	1 day
Factor for cyclic interval (3-255)	3 - 255 in single steps, **10** default setting

When "Functional selection" has been set to "Impulse counter" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Impulse counter

Input X	
Parameter	**Setting**
Count the impulses when	**rising edge**
	falling edge
	rising and falling edge
Base for cyclic interval	0.1 seconds
	1 second
	1 minute
	1 hour
	1 day
Factor for cyclic interval (3-255)	3 - 255 in single steps, **10** default setting

When "Functional selection" has been set to "Switching counter" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Switching counter
Count the impulses when	**rising edge**
	falling edge
	rising and falling edge
Command when maximum counter reading reached	sends no telegram
	sends ON telegram
	sends OFF telegram
	toggles
Maximum counter reading (1-65535)	1 - 65535 in single steps, **65535** default setting
Gradual counter output (1-255)	1 - 255 in single steps, **10** default setting

When "Functional selection" has been set to "Reset counter" in the "Input X" tab:

Input X	
Parameter	**Setting**
Functional selection	Reset counter
Affects input (1-4)	1 - 4
Behaviour on rising pulse edge	no action
	sends ON telegram
	sends OFF telegram
	toggles
Behaviour on falling pulse edge	no action
	sends ON telegram
	sends OFF telegram
	toggles
Counter and cycle time	do not reset
	reset on rising pulse edge
	reset on falling pulse edge
	reset on rising and falling pulse edge

Input X	
Parameter	**Setting**
Functional selection	Cyclical monitoring
Operating mode	make contact
	break contact
Number of objects	one
	two

merten

Multi-function.counter 12BA/1.0

Input X - (object A/B)	
Parameter	**Setting**
Object A/B	1 bit
	2 bit (priority control)
	4 bit
	1 byte in steps 0% - 100%
	1 byte infinitely 0 - 255
Action when activated	**transmits value 1 immediately and then cyclically**
	sends value 2 immediately and then cyclically
Action on release	**transmits value 1 immediately and then cyclically**
	sends value 2 immediately and then cyclically
value 1 / value 2	0 -100 % in 10% steps **100 %** default setting value 1, **0 %** default setting value 2
value 1 / value 2	0- 255, **255** default setting value 1, **0** default setting value 2
Base for cyclic interval	**0.1 seconds**
	1 second
	1 minute
	1 hour
	1 day
Factor for cyclic interval (3-255)	3 - 255 in single steps, **5** default setting

Input disable function	
Parameter	**Setting**
Disable function	**disabled**
	enabled
Block	For object value "0"
	For object value "1"
Behaviour at start of blocking	separately for each input
	all inputs function as a master input
Master input	Input **1**... 4

Inputs 1-X: Disable	
Parameter	**Setting**
Input X	**lock**
	do not disable
Cyclical actions are	**interrupted**
	interrupted and resumed after the blocking
	continued