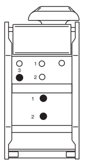
de en

merten

KNX Logic module Basic REG-K

Operating instructions



Art. no. 676090

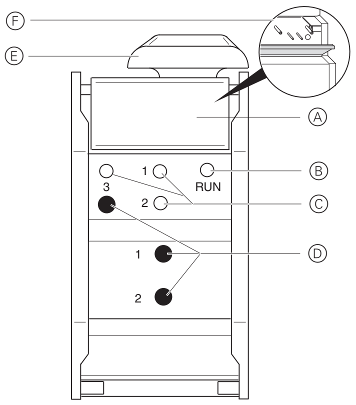
Getting to know the module

The KNX logic module Basic REG-K(called **module** in the following) facilitates control and regulation tasks. The received bus telegrams are interpreted and processed according to the programmable logic functions.

It is programmed via ETS.

For installation on EN 60715 DIN rails.

Connections, displays and operating elements



Ⓐ Flap, open forward

Ⓑ Operational LED (green): RUN

Ⓒ Channel LED (yellow) 1-3 for Ⓓ

Ⓓ Function keys 1-3

Ⓔ Cable cover

Ⓕ Behind the flap: Bus connecting terminal, programming button and programming LED (red)

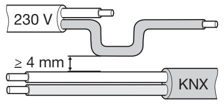
The 3 function keys have no function when they come from the factory. The keys and their functions first have to be enabled in ETS.

Mounting the module

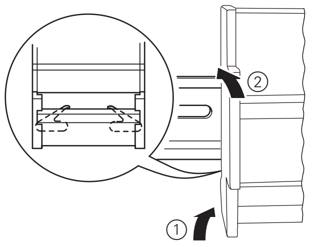
WARNING

Risk of fatal injury from electrical current. The device can be damaged.

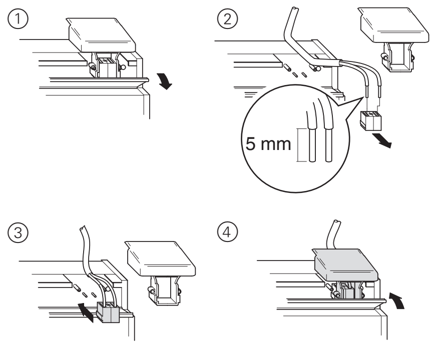
Ensure the safety distance per IEC 60664-1. There must be at least 4 mm between the individual cores of the 230 V supply cable and the KNX cable.



① Insert the module into the DIN rail with the clamping spring facing down and suspend it in the rail.



② Connect KNX.



③ Switch on the bus voltage.

④ Wait for at least 30 s.

Commissioning the module

① Press the programming button: the programming LED lights up.

② Load the physical address and application into the device from the ETS.

The programming LED goes out.

The operating LED lights up: the application has been loaded successfully and the device is operative.

Operating the module

① Enable the ETS push-button operation.

② Program each key with a logic function (links, disable and time functions)

③ Load the application into the device.

④ Press the key that you have programmed: the associated channel LED lights up when it is pressed.

The channel keys on the device are only there to retrieve logic functions without ETS.

Status LED

Operational LED (green)	**Programming LED** (red)	**Channel LED** (yellow)	
-	ON	-	The application is being loaded
On	-	-	Normal operation
On	-	On	Channel with function enabled in ETS

Operational LED (green)	**Programming LED** (red)	**Channel LED** (yellow)	
On	-	-	Normal operation
Off	-	-	No bus voltage
On	-	Lights up when activated	Channel with function enabled in ETS

Technical data

Power supply:	Via KNX DC 24 V, max. 16 mA
Operating elements:	1 programming key
	1 manual operation key
	1 channel key per channel
Display elements:	1 LED (red): Programming
	1 LED (green): RUN
	1 LED (red): Manual operation
	1 LED (yellow) per channel: Status
Ambient temperature:	
Operation:	-5 °C to +45 °C
Environment:	Can be used at elevations up to 2000 m above sea level (MSL)
KNX connection:	Two 1 mm pins for bus connecting terminal
Device width:	45 x 102 x 65 mm (W x H x D)

Merten GmbH

Merten GmbH, Solutions for intelligent buildings, Service Center, Fritz-Kotz-Str. 8, Industriegebiet Bomig-West, D-51674 Wiehl

Phone:	+49 2261 702-204
Fax:	+49 2261 702-136
E-Mail:	servicecenter@merten.de
Internet:	www.merten.com

If you have technical questions, please contact our InfoLine:

Phone:	+49 1805 212581* or +49 800 63783640
Telefax:	+49 1805 212582* or +49 800 63783630
E-Mail:	infoline@merten.de

*fee required

Application 7240/1.0 for Basic REG-K logic module

General

The KNX Basic REG-K logic module (art. no. 676090) can be programmed with this application.

In complex KNX installations, the logic module serves to establish special logic operations between sensors and actuators.

The logic module is a DIN rail mounted device for installing distributors. The connection to the KNX is established via the bus connecting terminal. An additional supply voltage is not required.

This application offers a wide range of possible settings for executing numerous logic functions for controlled KNX devices (e.g. dimming or switch actuators etc). Of course, which function is possible in each individual case depends on the KNX devices being controlled. In the following, only the KNX control functions and the objects relevant to these and parameters of the logic module are described. Due to the large number of possible settings, the logic module is particularly well suited to the areas of security, comfort or energy saving. The logic module serves solely to utilise bus telegrams. Only one application program is used for all settings.

i Configurable times (staircase timer, ON delay, OFF delay, etc.) are set via the time base and time factor parameters. The actual time is calculated by multiplying both values; e.g. base 1 second multiplied by factor 3 equals 3 seconds.

i If you load the logic module into your project via the ETS, all functions (in the "General" tab) are deactivated. Activate the function(s) you require.

Functions

If you load the logic module into your project via the ETS, all functions (in the "General" tab) are deactivated. Activate the function(s) you require.

The following functions can be selected:

Function	Number of blocks	Number of objects	Number of function objects
Logic	10	10	100
Time delay and filter	10	3	30
Converter	8	3	24
Multiplexer	12	4	48

– Total number of function objects: **202**
– Global objects: **6**
– Additional objects: for 3 push-buttons **and** 3 LEDs
– max. 230 objects
– max. 255 connections

i The setting examples shown in this application description serve merely as guidance and may deviate from the settings actually required.

i The **bold** values are the values set during factory configuration.

i The first block of a function is described in each case, since all blocks have the same parameters and setting values.

i Always set all parameters on the first block before parameterising the next block.

Behaviour after ETS application download

Downloading the application deletes all data required for the behaviour when the bus voltage is re-established. All input values are set to "0". Even if the "Status before bus voltage failure" setting is activated, the inputs are "0" after downloading. Likewise, the gate is always closed. This means that the settings for the behaviour when bus voltage fails do not apply to the download.

General parameters

A few parameters that are relevant to all functions and their settings, and consequently also their behaviour, will be described first, before the individual functions of the logic module are elaborated on.

These parameters are as follows:

• Behaviour when bus voltage is re-established
• Gate function
• Internal connection

Behaviour when bus voltage is re-established

Start-up delay

Time delay between re-establishment of the bus voltage and the functional start of the logic module.

Set a time from which the reading of the input telegram is successful.

Parameter name	**Objects**
Start-up delay after re-establishment of the bus voltage in s	1 ... 120, **25**

Input objects

General input objects: Logic object, time delay and filter object, converter object and multiplexer object.

Input behaviour

The behaviour of the input after re-establishment of the bus voltage can be determined here.

Reads the current value: A status inquiry is sent to the bus and the response is awaited. The inquiry is repeated every minute until the first telegram arrives at the input. For this setting, it is essential that the read flag (R-flag) is set for the corresponding sensor or actuator. During commissioning (perform reset), check that the read operation was completed successfully and that re-reading every minute following successful receipt was set.

An unnecessary burden is placed on the bus if the L flag is not set (too many cyclical telegrams) and the other bus functions are impaired to too great an extent.

Waiting for new telegram: No inquiry is sent to the bus. The input waits for the first new telegram.

Status before bus voltage failure: All inputs are replaced with the values stored in the memory after the bus voltage is re-established.

0 until first telegram: The value of the input object is "0" until another telegram (apart from 0telegrams) is received.

1 until first telegram: The value of the input object is "1" until another telegram (apart from 1telegrams) is received.

Gate function

All function blocks listed above include a gate function at the output with which the output behaviour can be adjusted.

The gate and its adjustable parameters are the same for all functions of the logic module that can be selected in the ETS.

Input behaviour

The gate is either open (all telegrams are let through) or closed (no telegram is let through). The behaviour can be inverted.

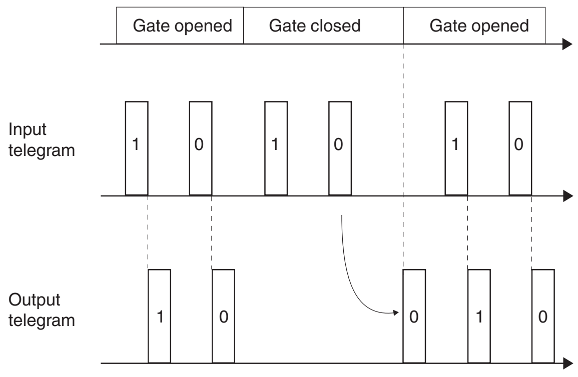
Gate behaviour

The gate has either the value 1 or 0. The behaviour can also be inverted.

Output behaviour

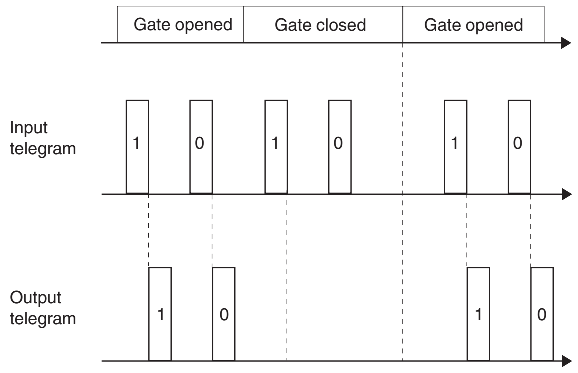
The user can select whether the gate sends a telegram upon opening or not and whether the value of the output is inverted.

Example 1:



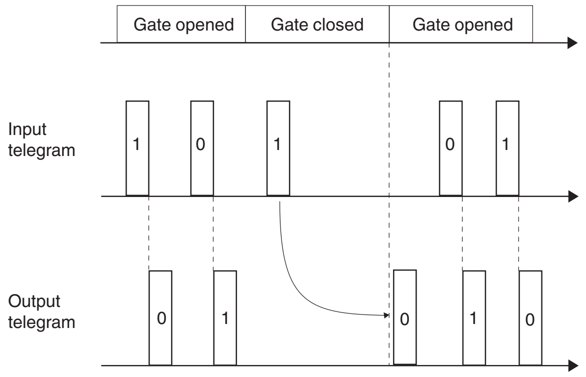
The gate closes and the last incoming telegram is saved. The gate is opened again and the saved telegram is forwarded.

Example 2:



Setting: The gate closes, no telegram is saved. The gate is opened again and the first incoming telegram is forwarded.

Example 3:



Setting: The gate closes, the last telegram is saved. The status of the output is inverted. The gate is opened again and the saved telegram is forwarded with inverted value (1 -> 0).

Internal connection

The internal connection function serves to reduce the number of group addresses and telegrams, thereby considerably reducing the bus load.

The internal connection can be activated for the logic function, the time delay and filter function, as well as for the converter function. The adjustable connections are always the same. The logic module only ever supports the additional "internal connection" function for the first input of a function block. Other functions are supported via a virtual input, which can be selected freely.

Special effects are achieved by combining internal connections and group addresses (e.g. complex logic connections or block connections without group address).

Double assignment (internal **and** with a group address) should only be implemented in justified exceptional cases.

The modules are worked through in the following order:

- Logic function
- Time delay and filter function
- Converter function
- Multiplexer function

Furthermore, the blocks of each function are worked through one after the other (first logic block 1, then logic block 2 through to logic block 10. Subsequently, time delay and filter block 1, etc.).

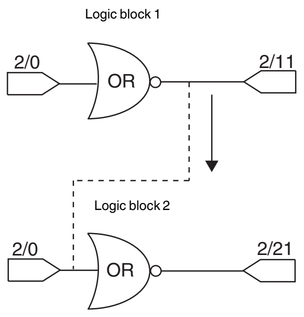
If an input is connected to both an internal connection and also to a group address, the result at the output depends on whether the internal connection comes from a "higher" or "lower" block.

Examples

- Below is described how a logical AND operation can be established between two internal connections.

Example 1: Block 1 and block 2 invert the input values in each case. The output from block 1 is internally connected to the input from block 2.

Block 1 and 2 are updated simultaneously. First, block 1 is recalculated, which then alters the input for block 2. Block 2 now has a new input value. Both steps are completed in one cycle.

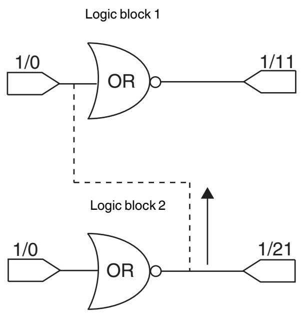


--- Internal output connection

The input telegram (2/0) has the value 0. The output telegrams have the values 1 (2/11) and 0 (2/21). This means that the input value "0" is overwritten with a "1" by the internal connection at the input to block 2.

Example 2: Block 1 and block 2 invert the input values in each case. The output from block 2 is internally connected to the input from block 1.

Block 1 and 2 are updated simultaneously. First, block 1 is recalculated. In the next stage, block 2 is calculated. The result from block 2 updates the input to block 1. In the next cycle, the value of block 1 is recalculated. The output value for the group address 1/11 was altered twice. The time delay between the two results depends on the number of blocks used.



--- Internal output connection

Logic function

Object name	Size	Flag	Direction
Logic object 1...8	1 bit	CW	Input
Logic gate input control	1 bit	CW	Input
Logic output	1 bit	CT	Output

Logic operation

There are a total of 10 logic blocks available.

You can choose from one of the following logic gates for each logic block: **AND / OR / EXCLUSIVE OR (XOR)**. All gates can be inverted.

A B	OR	NOR
0 0	0	1
0 1	1	0
1 0	1	0
1 1	1	0

A B	AND	NAND
0 0	0	1
0 1	0	1
1 0	0	1
1 1	1	0

A B	XOR	XNOR
0 0	0	1
0 1	1	0
1 0	1	0
1 1	0	1

The difference between the "or" and "exclusive or" logic operations is that the output from the XOR gate is logical "1" if and only if there is an unequal number of "1" and "0" inputs. In the simple case of an XOR gate with two inputs, this means that the inputs must be different to one another to obtain the output "1". "1" must be present at precisely one of the two inputs.
In contrast to a simple OR logic operation, the condition is deemed not to be met if a "1" is present at both inputs. With an XOR gate, the result in this case is a "0". Each additional input at the gate alters the behaviour accordingly.

A B	OR	XOR
0 0	0	0
0 1	1	1
1 0	1	1
1 1	1	0

A B C	OR	XOR
0 0 0	0	0
0 0 1	1	1
0 1 0	1	1
0 1 1	1	1
1 0 0	1	1
1 0 1	1	1
1 1 0	1	1
1 1 1	1	0

Input behaviour

The input telegrams can be inverted for each input. In addition, a fixed value (0 or 1) can be assigned.

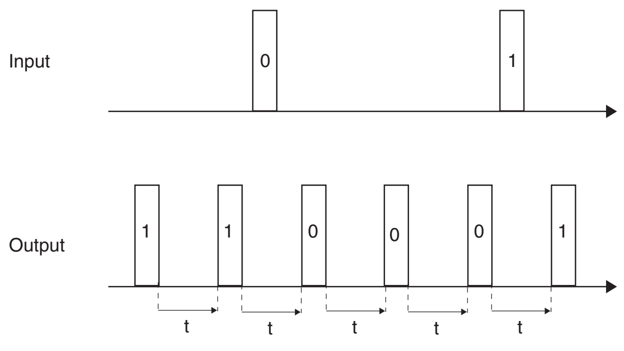
Output behaviour

Criteria for the sending behaviour at the output can be defined.

Output change: A telegram will only be sent if the result of the logic operation alters. This means that cyclical input telegrams at the output do not trigger cyclical telegrams.

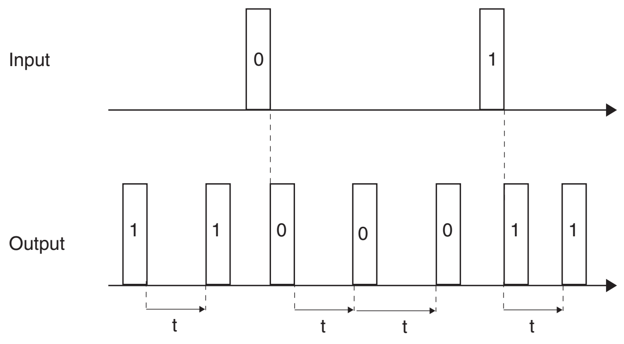
Receipt of an input telegram: An output telegram will be sent after receipt of an input telegram, regardless of the logic result. This means that cyclical input telegrams also trigger cyclical output telegrams (same cyclic interval).

Cyclically: An output telegram will be sent exclusively at the set cyclic intervals. This cyclic interval consists of a selectable basis (1 s or 1 min) and an additional adjustable factor (1...65535). This means that non-cyclical input telegrams are also converted into cyclical output telegrams.



t = Cyclic interval

Cyclically and output change: In addition to the cyclical sending, output telegrams will also be sent in the case of changes at the input. The send conditions for output telegrams outside the cyclic interval result from the definition of send criteria (output change or receipt of a telegram). This setting is wise if cyclical telegrams and a rapid response are expected (e.g. weather alarm at the blind actuator).



t = Cyclic interval

Basic applications

The logic function is particularly suitable for summarising messages (e.g. the lighting status in rooms), linking conditions (e.g. rain or wind sensor activates a safety function) or programming an additional toggle between manual and automatic (e.g. disabling brightness-dependent lighting control for a video presentation).

Parameter

Logic block 1	
Parameter	**Setting**
Logic gate	**AND** OR EXCLUSIVE OR
Logical inputs	
Value of logic object 1	**Use** inverted
Behaviour of logic object 1 when bus voltage re-established	**Reads current value** Waits for new telegram Status before bus voltage failure until first telegram 1 until first telegram
Value of logic object 2	**Do not use** use inverted =1 =0
Value of logic object 3	**Do not use** use inverted =1 =0
Value of logic object 4	**Do not use** use inverted =1 =0
Value of logic object 5	**Do not use** use inverted =1 =0
Value of logic object 6	**Do not use** use inverted =1 =0
Value of logic object 7	**Do not use** use inverted =1 =0
Value of logic object 8	**Do not use** use inverted =1 =0
Parameter	**Setting**
Gate function	**1= closed, 0= opened** 0 = closed, 1 = opened (inverted)
Send output telegram if gate opens	Yes **No**
Behaviour of gate when bus voltage re-established	**Reads current value** Waits for new telegram Status before bus voltage failure Opened Closed
Value of output objects	**uses** inverted
Send result	**After output change** After receiving an input telegram Cyclically Cyclically and after output change
Cyclic interval = basis x factor	
Basis	1 s **1 min**
Factor	1 ... 65535, **10**
Internal connection	**Yes** **No**

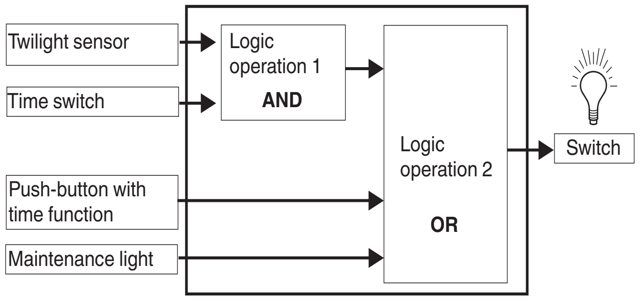
The value of logic object 1 is either "use" or "inverted". It is not possible to set a fixed value or "do not use". The value of logic object 2...8 can be "do not use", "use", "inverted", =0 or =1.

Internal connection	
Parameter	**Setting**
Output of basic logic function logic block 1 is connected to	**Nothing** Logic block 1 logic object 1 ... Logic block 10 logic object 1 Time and filter block 1 ... Time and filter block 10 Converter block 1 ... Converter block 8 Signal 1 Signal 2 Signal 3

Never connect the output and the input of the same logic blockto one another, as this can cause the device to malfunction.

Application example

- A light-sensitive switch switches the lighting on automatically.
- The light is switched off between 23:00 and 06:00.
- In the morning, the light switches on from 06:00 when it is dark.
- In addition, the light can be switched on for 5 minutes at any time via a push-button.
- A continuous light function is possible for maintenance purposes.



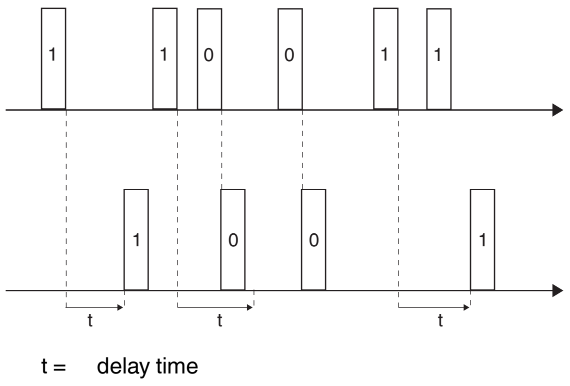
Time delay and filter function

Object name	Size	Flag	Direction
Time delay and filter object	1 bit	CW	Input
Gate input control filter	1 bit	CW	Input
Time delay and filter output	1 bit	CT	Output

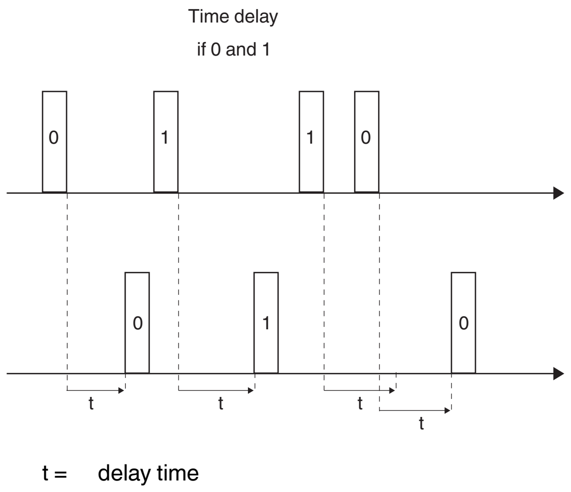
Time delay

Output telegrams can be sent with a time delay. Switch-on and/or switch-off times can be adjusted depending on the input telegram. The time delay can also be deactivated.

Example: **Time delay if 1**. The 1telegram is forwarded with a time delay. The 0telegram cancels the time delay.



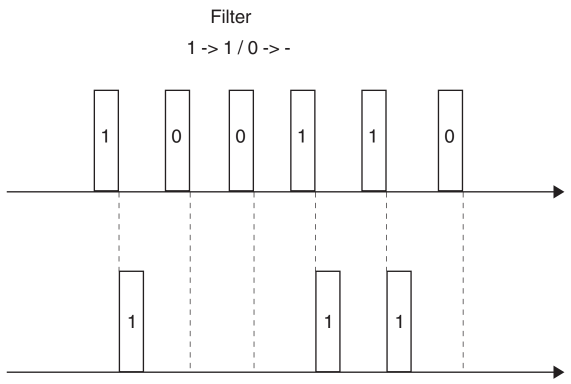
Example: **Time delay if 0 and 1**. Both telegrams are forwarded with a time delay.



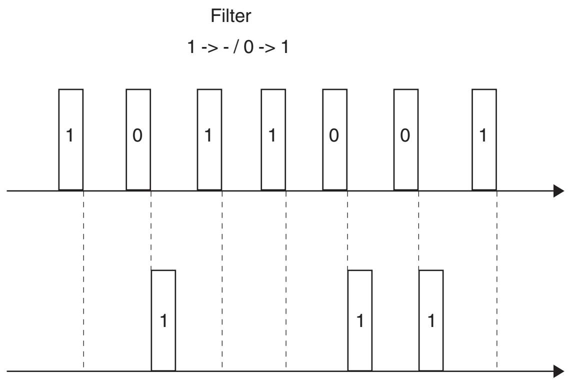
Filter function

Ten different assignments of input and output telegrams are available. It is possible to switch ON or OFF or to TOGGLE, to send only certain telegrams (e.g. ON-> ON, OFF -> -) or to invert the input value.

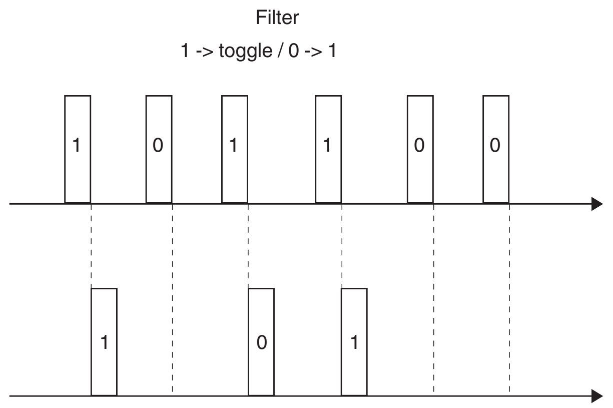
Example 1: **1 -> 1 / 0 -> -**. 1telegrams are let through and 0telegrams are filtered out.



Example 2: **1 -> - / 0 -> 1**. 1telegrams are filtered out and 0telegrams are converted into 1telegrams.



Example 3: **1 -> toggle / 0 -> -**. 0telegrams are filtered out. The 1telegrams toggle between 0 and 1.



10 blocks are available for time delay and filter functions. All blocks are deactivated and must be activated individually and functions must be assigned to them.

Basic applications

The time delay function and filter are particularly suitable for sending messages with a time delay (e.g. darkening the building after closing the windows or comfort extension), adapting signals (e.g. key is not able to suppress 1, 1 is filtered out). Alternatively, toggling between manual and automatic can be programmed.

Parameter

Time delay and filter function	
Parameter	**Setting**

Time delay	**No**
	if 1
	if 0
	if 0 and 1

Filter	**1 -> 1 / /0 -> -** 1 -> - / 0 -> 0 1 -> 1 / 0 -> 0 1 -> - / 0 -> - (switched off) 1 -> - / 0 -> 1 1 -> 0 / 0 -> - 1 -> 0 / 0 -> 1 1 -> toggle/ 0 -> - 1 -> - / 0 -> toggle 1 -> toggle 0 -> toggle
Status of the time and filter object after bus voltage re-established	Reads current value **Waits for a new telegram** Status before bus voltage failure 0 until first telegram 1 until first telegram
Gate function	**1= closed, 0 = open** 0 = closed, 1 = open (inverted)
Send output telegram if gate opens	Yes **No**
Behaviour of gate when bus voltage re-established	Reads current value **Waits for new telegram** Status before bus voltage failure opened closed
Send result	After output change **After receiving an input telegram** Cyclically Cyclically and after output change
Use internal connection	Yes **No**

Internal connection	
Parameter	**Setting**
Time delay output and block 1 filter function is connected to	**Nothing** Logic block 1 logic object 1 ... Logic block 10 logic object 1 Time and filter block 1 ... Time and filter block 10 Converter block 1 ... Converter block 8 Signal 1 Signal 2 Signal 3

Never connect the output and the input of the same block to one another (internal connection or group addresses), as this can cause the device to malfunction.

Converter function

Object name	Size	Flag	Direction
Converter input/ output object	1 bit 2 bit 1 byte	CW CT CT	Input Output Output
Converter gate input control	1 bit	CW	Input

1 bit signals can be converted into 2 bit or into 1 byte signals and 1 byte signals can be converted into 1 bit signals using the converter function.

There are 8 converter blocks available. All blocks are deactivated and must be activated individually and functions must be assigned to them. The transition from "0" to "1" can be adjusted.

Basic applications

1 bit -> 2 bit conversion: Switching with priority, e.g. load management.

1 bit -> 1 byte conversion: Limit value with 1 bit is used to retrieve a lightscene.

1 byte -> 1 bit conversion: 1 byte value generates 1 bit status feedback for an LED.

Parameter

Converter block 1	
Parameter	**Setting**
Converter function	1 bit -> 2 bit / **1 bit <-> 1 byte**
Value for 0telegrams	**0** ... 255
Value for 1telegrams	0 ... **255**
0telegram is generated if 1 byte value is < ...	0...255, **1**
Status of the converter after re-establishment of the bus voltage	Reads current value **Waits for new telegram** Status before bus voltage failure 0 until first telegram 1 until first telegram
Send output telegram if gate opens	Yes **No**
Send result	**After output change** After receiving an input telegram Cyclically Cyclically and after output change
Use internal connection	Yes **No**

Internal connection	
Parameter	**Setting**
Output converter function block 1 is connected to	**Nothing** Logic block 1 logic object 1 ... Logic block 10 logic object 1 Time and filter block 1 ... Time and filter block 10 Converter block 1 ... Converter block 8 Signal 1 Signal 2 Signal 3

i Never connect the output and the input of the same block to one another (internal connection or group addresses), as this can cause the device to malfunction.

Example 1 bit -> 2 bit

- A room is controlled via KNX.
- In the event of fire, 1/3 of the complete lighting should be switched on.
- The selected priority control ensures that this target is achieved.

Example 1 byte -> 1 bit

- The heating control is monitored by a visualisation.
- An LED display appears in the visualisation when the valve position is exceeded by x%.
- Monitoring of room temperature and central reduction of the setpoint temperature of the heater, where necessary.

Multiplexer function

Object name	Size	Flag	Direction
Multiplexer Input / output object A, B	1 bit 2 bit 4 bit 1 byte 2 byte 4 byte	CWT	Input / output
Control object	1 bit	CW	Input
Multiplexer gate input control	1 bit	CW	Input

The gate integrated into the multiplexer serves to control the flow of data.

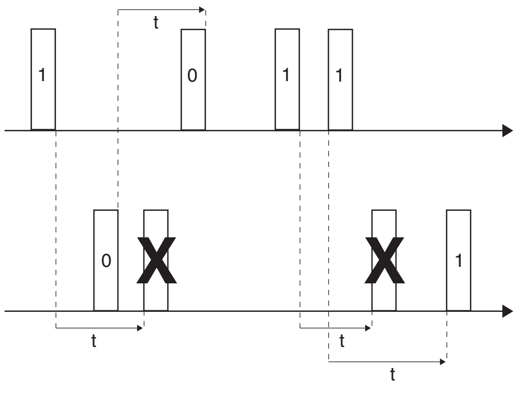
The following formats can be selected:

– 1 bit
– 2 bit
– 4 bit
– 1 byte
– 2 byte
– 4 byte (only in the first multiplexer block)

The multiplexer is bidirectional and the data direction can be altered via the control object.

There are 12 multiplexer blocks available. All blocks are deactivated and must be activated individually and functions must be assigned to them.

The time delay is set separately for each individual multiplexer block. The time delay can be retriggered after a new update is received.

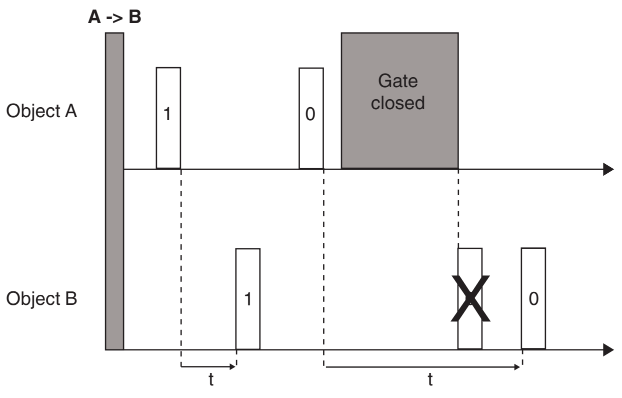


t = time delay (adjustable)

Output behaviour

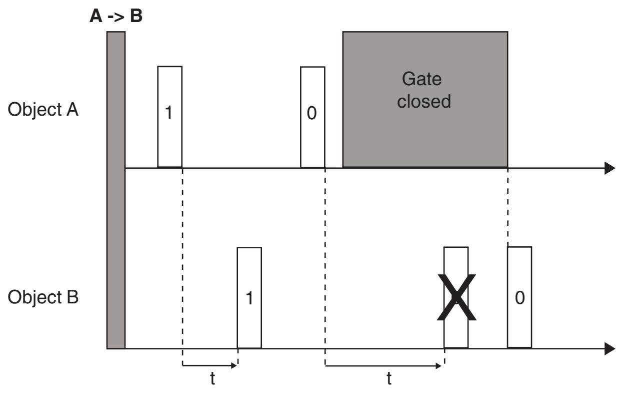
Send output telegram if gate opens: An output telegram is sent after the gate status changes (gate is opened). This only happens once the delay time has elapsed, however. The telegram is not sent immediately after the gate opens, but instead only once the delay time has elapsed. If the delay time has already elapsed before the gate opens, however, the telegram will be sent immediately when the gate is opened.

Example 1: The gate is closed and the delay time is active. The gate is open and the time has not yet elapsed. An output telegram will be sent when the delay time has elapsed.



t = delay time

Example 2: The gate is closed and the delay time is active. The time has elapsed and the gate is still closed. If the gate is opened, an output telegram will be sent immediately.



t = delay time

Basic applications

The multiplexer is particularly suitable for controlling conference rooms (e.g. a large conference room can be divided into several small rooms using moveable walls. Push-button signals are then only forwarded to the respective sections).

Parameter

Multiplexer block 1	
Parameter	**Setting**
Type of multiplexer object	**1 bit** 2 bit 4 bit 1 byte 2 byte 4 byte
Control object = "0"	A / B A -> B A <- B **A <-> B**

Control object = "1"	**A / B**
	A -> B
	A <- B
	A <-> B

Gate function	**1= closed, 0 = opened** 0 = closed, 1 = opened (inverted)
Send output telegram if gate opens	**Yes** **No**
Behaviour of gate when bus voltage re-established	Reads current value **Waits for a new telegram** Status before bus voltage failure 0 until first telegram 1 until first telegram
Send result	**After output change** After receiving an input telegram Cyclically Cyclically and after output change
Factor: Output telegram delay (basis = 10 ms; 0 = no delay)	**0** .. 65535

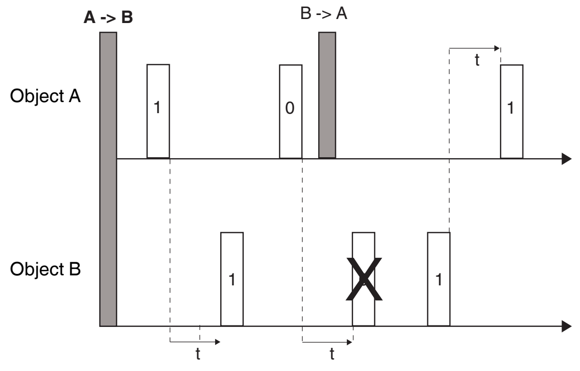
Internal connection	
Parameter	**Setting**
Output multiplexer function block 1 is connected to	**Nothing** Logic block 1 logic object 1 ... Logic block 10 logic object 1 Time and filter block 1 ... Time and filter block 10 Converter block 1 ... Converter block 8 Signal 1 Signal 2 Signal 3

Do **not** connect the output from **multiplexer block 1** to **multiplexer block 1**, as this can cause the device to malfunction.

Control object

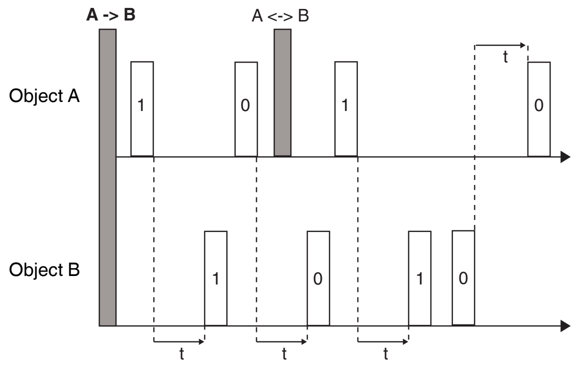
Control object **A -> B** specifies the direction and the value telegram is delayed.

Example 1: The multiplexer deletes the previous telegram following the change of direction to **B -> A**, since **B** is the input object this time.



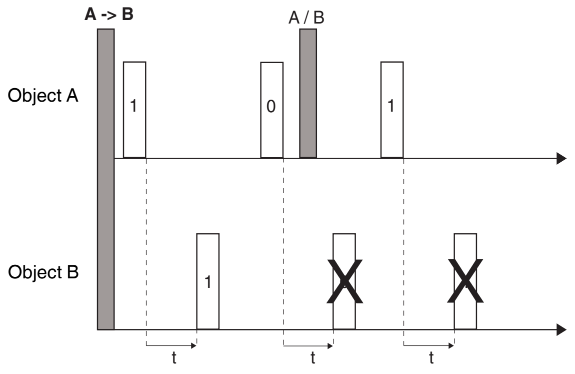
t = delay time

Example 2: The previous telegram is sent to the bus following the change of direction to **A <-> B**. Both objects are the input and the output simultaneously.



t = delay time

Example 3: The previous telegram is deleted following the change of direction to **B / A**, as the multiplexer block is deactivated.



t = delay time

Channel LEDs and channel push-buttons

Each individual push-button and each individual LED can be assigned its own function. These possibilities are particularly suitable for testing (e.g. sending input telegrams to a logic object at the push of a button) or checking (e.g. LED lights up when logic function is activated) logic functions. Furthermore, push-buttons and LEDs connected to the push-button input objects can temporarily switch off connected logic operations.

Parameter	Size	Flags	Direction
LED 1, LED 2, LED 3	1 bit	CW	Input
Push-button 1, push-button 2, push-button 3	1 bit	CT	Output

Channel push-buttons and channel LEDs	
Parameter	**Setting**
Behaviour of LED 1 after receiving the signal 1 = 1/0	**ON / OFF** OFF / ON Flash / OFF OFF / Flash Flash / ON ON / Flash always off
Status of push-button object 1 after push-button 1 was pressed	**Toggle / -** 1 / 0 0 / 1 1 / - 0 / - Deactivated
Use internal connection	Yes / **No**
Behaviour of LED 2 after receiving the signal 2 = 1/0	**ON / OFF** OFF / ON Flash / OFF OFF / Flash Flash / ON ON / Flash always off
Status of push-button object 2 after push-button 2 was pressed	**Toggle / -** 1 / 0 0 / 1 1 / - 0 / - Deactivated
Use internal connection	Yes / **No**
Behaviour of LED 3 after receiving the signal 3 = 1/0	**ON / OFF** OFF / ON Flash / OFF OFF / Flash Flash / ON ON / Flash always off
Status of push-button object 3 after push-button 3 was pressed	**Toggle / -** 1 / 0 0 / 1 1 / - 0 / - Deactivated
Use internal connection	Yes **No**
Minimum switching time factor (basis = 0.5 s)	1 ... 255

The minimum switching time factor is the minimum time that a key must be pressed to send a switching signal to the bus.

Schneider Electric Industries SAS

If you have technical questions, please contact the Customer Care Center in your country.

www.schneider-electric.com

This product must be installed, connected and used in compliance with prevailing standards and/or installation regulations. As standards, specifications and designs develop from time to time, always ask for confirmation of the information given in this publication.