# Technical Manual

## FANCOIL ACTUATOR SINGLE VALVE
## COOLING or HEATING
## ANALOG or PWM

Art. 1730.02120/55100

DATEC
ELECTRONIC AG
MEMBER KNX
Hard-&Firmware Engineering

**DATEC Electronic AG**
Hard-& Firmware Engineering
Hochbergerstrasse 60C, CH-4057 Basel
Tel. +41 (0) 61 633 22 25, Fax. +41 (0) 61 633 22 27
www.datec.ch, support@datec.ch

Because of continuous product improvement, notice is subject to change without notification. Please always check for latest version. Refer to www.datec.ch or contact support@datec.ch

# 1. Overview

## *1.1. Presentation*

The ***KNX FANCOIL ACTUATOR SINGLE VALVE*** is a KNX device for making fancoils work together with other products using the KNX bus.
It acts as an interface between the KNX bus, the worldwide standard for home and building automation, and valve driven heating or cooling devices.

The fancoil actuator is able to drive:

- The fan, with up to 3 speeds
- The heating or cooling valve
- A relay

As physical inputs, the fancoil has:

- 1 input for connecting a temperature sensor
- 1 digital input, to be used as a binary input or window contact input

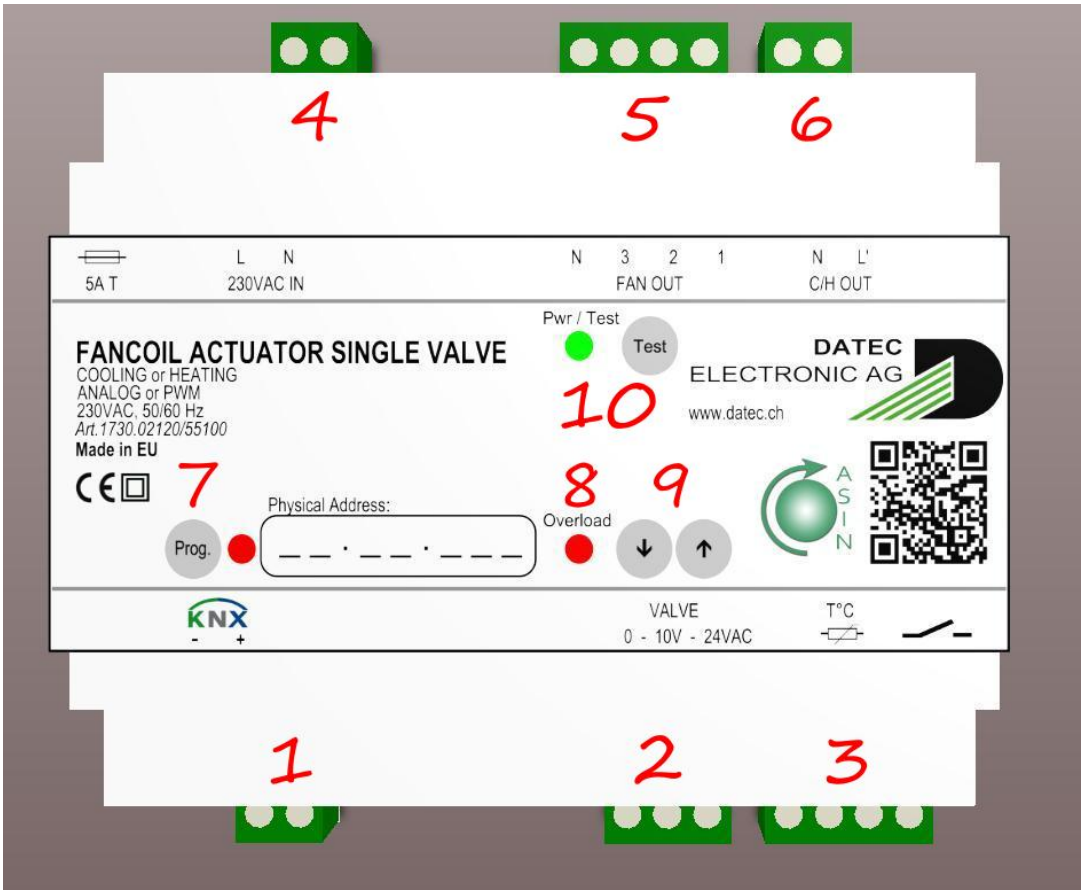
Together with a KNX room temperature controller, the device can be used for temperature regulation of offices, hotel rooms, …
The device also sends different status for displaying in application.
Complete description is made for cooling mode. If heating is chosen same description applies, simply replace “cooling” by “heating”.

To configure and operate the fancoil actuator, at least ETS version 4 is needed.

The product database is available for download: http://www.datec.ch/1730.02120-55100.html

**Because of high voltage at terminals, the device is to be installed by skilled personal only!**

## *1.2. Overall view*




**Terminals:**

1. KNX terminals
2. Valve terminals
3. Local temperature sensor & Binary input / Window contact input
4. 230VAC supply
5. 230VAC fan outputs
6. 230VAC relay output

**Buttons:**

7. KNX programming button
9. Increment and decrement buttons
10. Test button

**LED's:**

7. KNX programming LED
8. Overload LED
10. Pwr/Test LED

## *1.3. Connection diagram*

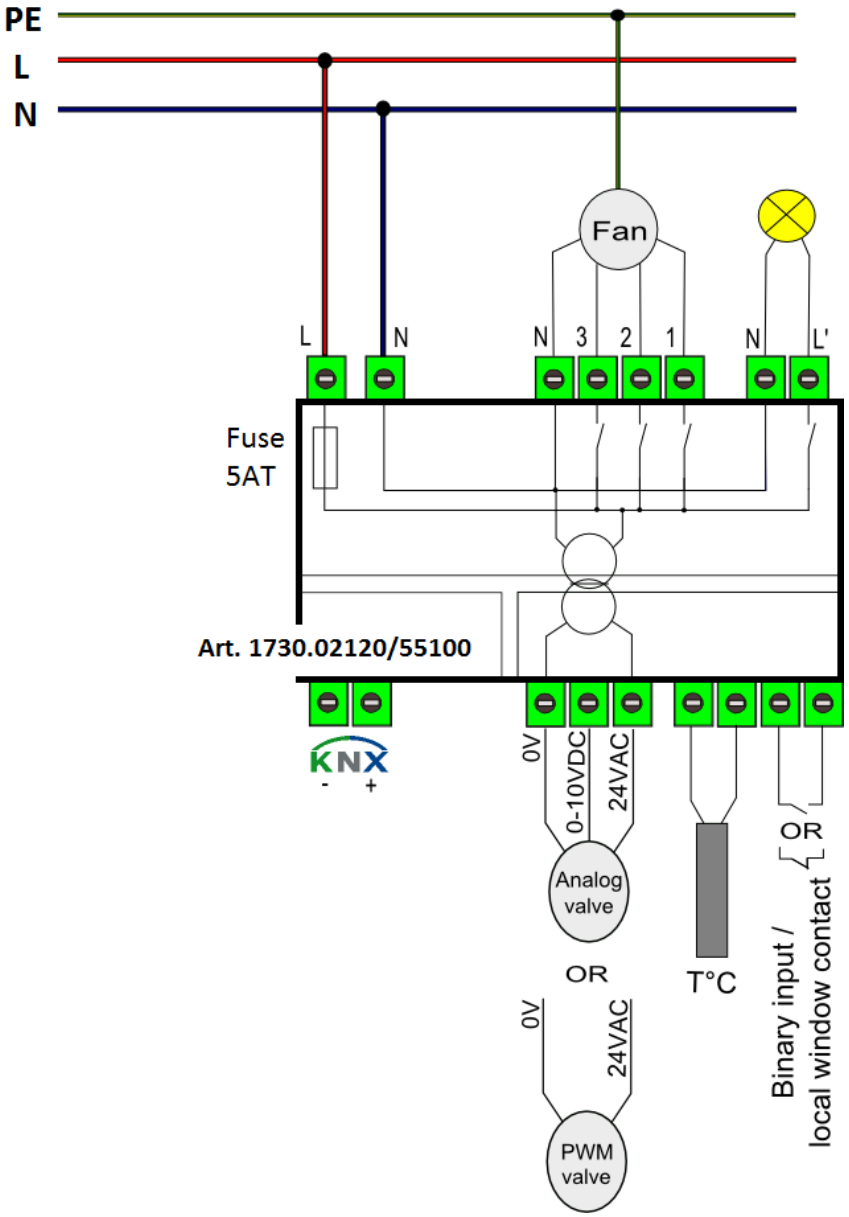



Figure 1: Connection diagram

# 2. Operating manual

## *2.1. Inputs / Outputs*

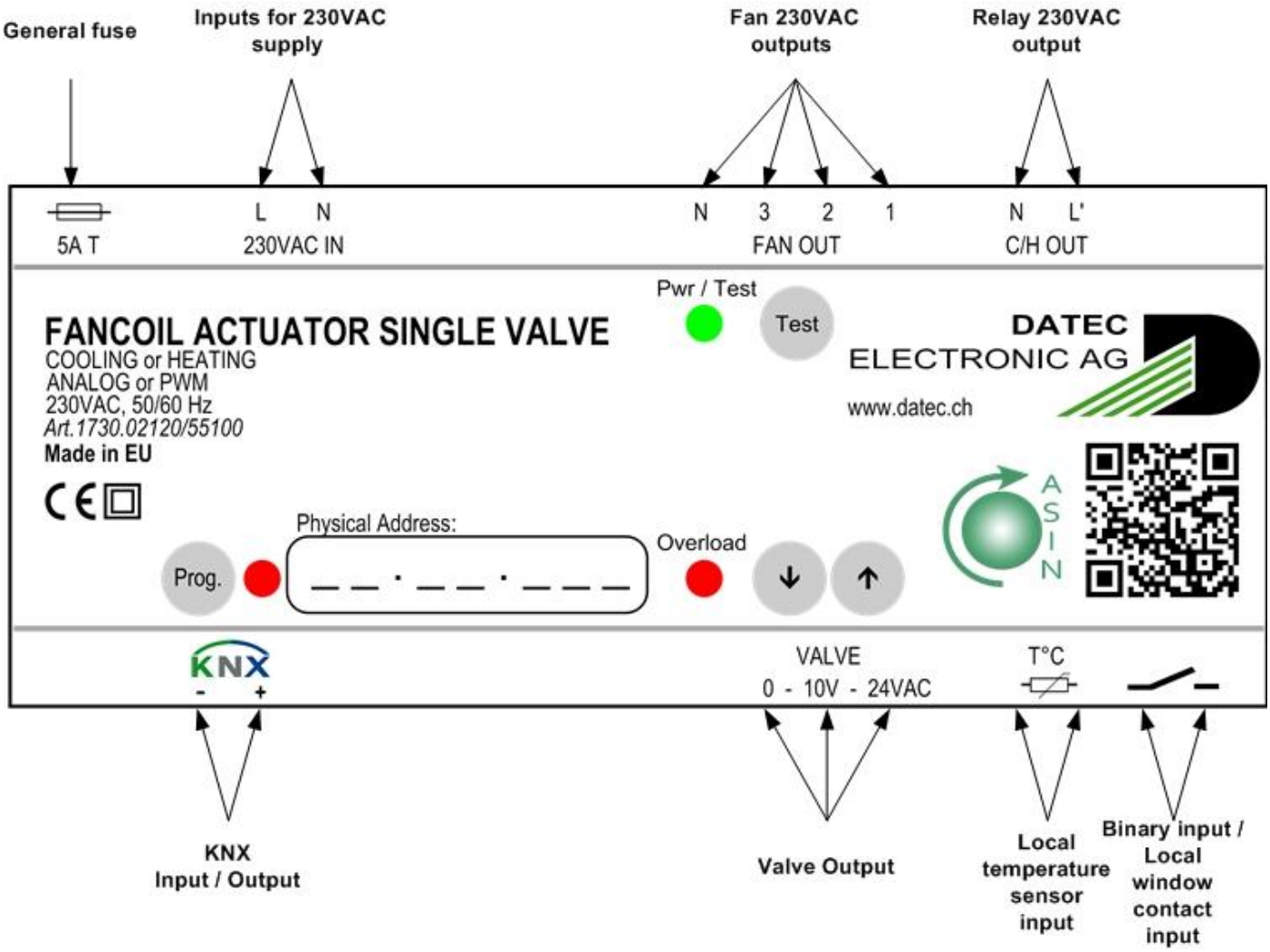



Figure 2: Inputs / Outputs

## 2.1.1 Power (input)

The device is powered by one single 230VAC input, protected by a general 5A T replaceable fuse.
This 230VAC is used for internal powering, but also for feeding the fan and the load on the relay output.
This 230VAC is internally transformed to isolated 24VAC for feeding the valve and to 24VDC for feeding the temperature sensor and the digital input / Window contact input.
In case of 230VAC power failure, the device will not work.

## 2.1.2 Fan (output)

The device is designed to drive a 230VAC fan with 1, 2 or 3 different fan speeds.
For each fan speed, one separate relay is switching 230VAC on the corresponding output terminal.

## 2.1.3 Relay (output)

The device has 1 more internal 230VAC relay, switching 230VAC on the corresponding output terminal.

## 2.1.4 KNX (input / output)

The device is connected to the KNX bus through 2 terminals.
The KNX bus is used for bi-directional communication with other devices.
In case of KNX failure (short circuit, missing bus, …) the device can still continue to work (temporary ***Handmode*** or Default mode after monitoring time elapsed), because powered by 230VAC.
The KNX is internally isolated from 230VAC, 24VAC and 24VDC.

## 2.1.5 Valve (output)

The device is designed to drive a single valve, either for heating or for cooling purpose.
Two types of valve can be used:

- Analog 0-10VDC:
  If using an analog 0-10VDC valve, the device is providing a 24VAC, overload protected, power supply to the valve.
  Moreover, a 0-10VDC modulated output signal is generated by the device, in order to drive the valve to the needed position.
  For closing the valve, the device can output a minimum signal of 0VDC.
  For opening the valve, the device can output a maximum signal of 10VDC.
  Voltages in between can be used for intermediate valve positions (255 steps of resolution)..
- 24VAC PWM:
  If using a 24VAC pulse width modulated valve, the device is providing a 24VAC, overload protected, pulse ratio modulated signal.
  For closing the valve, the device can output a minimum pulse ratio of 0% from 24VAC signal (always OFF).
  For opening the valve, the device can output a maximum pulse ratio of 100% from 24VAC signal (always ON).
  Rations in between can be used for intermediate valve positions (255 steps of resolution).
  The PWM Period can be set to several values between 15seconds and 1 hour.

Independently of the type of valve used, the output current on 24VAC is continuously monitored inside the device.
Short or small overloads are admitted. If small overloads are lasting too long, or if even short overloads are too high, the 24VAC supply of the valve is switched OFF for a given time, and will automatically recover.
After recovering, the same measurement takes place again.

So, overheating of device due to overloads on valve due to miss-wiring or short circuit can be reduced significantly.

The nominal power of valve should not exceed 4VA.
Connections to valve must be limited to maximum 3m.

## 2.1.6 Local temperature sensor (input)

A local temperature sensor can be connected to the device.
The measurement range is from 0.0°C to +50.0°C.
The measured temperature is to be sent on bus, to be used by displays or room temperature controller.
Only temperature sensors from DATEC Electronic AG are to be used.

Connection to local temperature sensor must be limited to maximum 30m. Only twisted pair has to be used.

## 2.1.7 Binary input / Local window contact (input)

A digital input can be connected to the device. This could be used as simple binary input or as local window contact.
The device is providing a 24VDC voltage on the corresponding terminals, so only dry contacts have to be used.
The maximum output current is limited to about 6mA.
The polarity of the external contact can be set by parameter.

Connection to local binary input / local window contact must be limited to maximum 30m. Only twisted pair has to be used.

## *2.2. Local push-buttons and LED's*

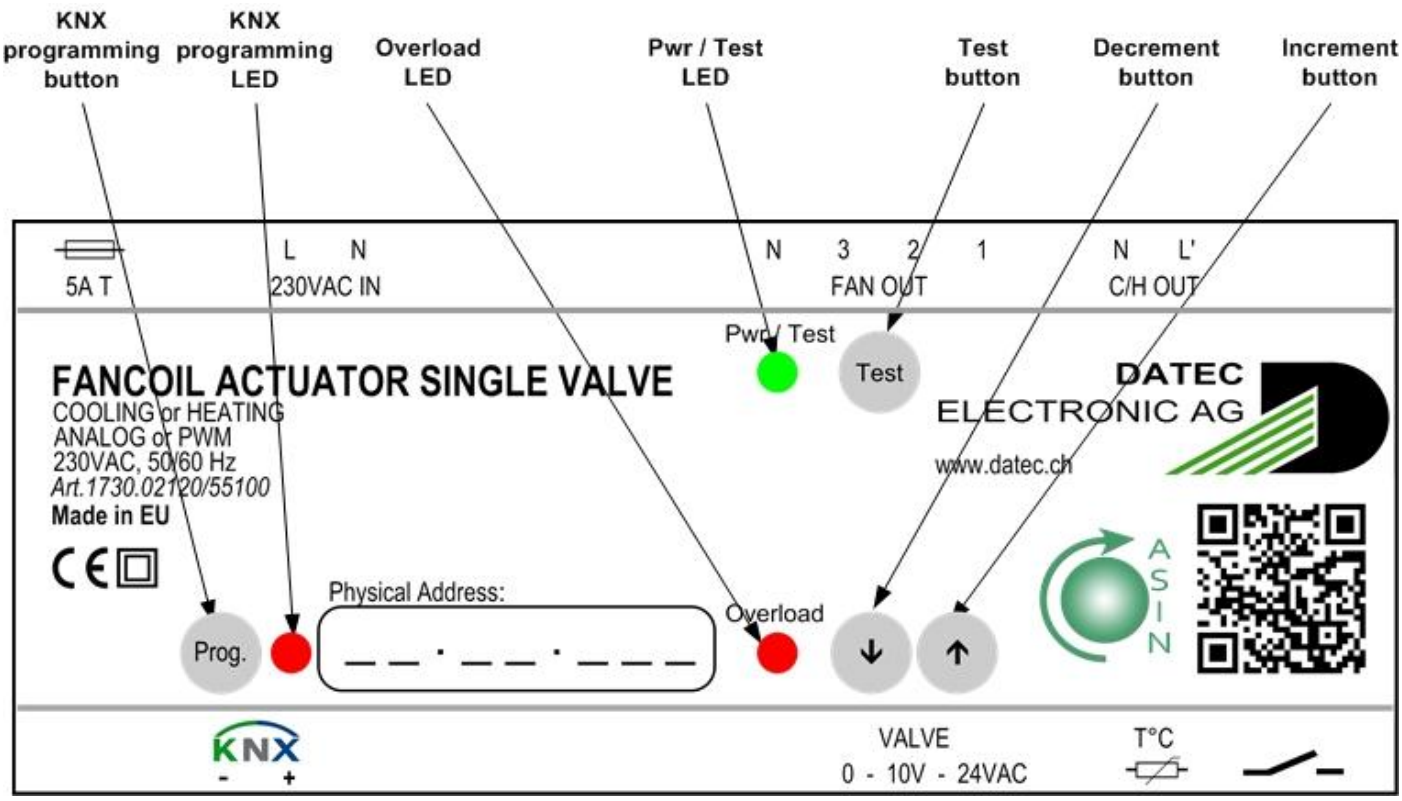



Figure 3: Local push-buttons and LED's

A temporary ***Handmode*** is available for testing purpose, in order to check installation even without device being programmed or connected to KNX bus.

***Handmode*** is deactivated if not in operating mode or after master reset!

### 2.2.1 KNX Programming button "Prog." and LED

Pressing the KNX programming button ***"Prog."*** will put the device in KNX Programming mode. The KNX programming mode will be exited automatically after successful physical address programming, resetting device or pressing again the ***"Prog."*** button.

The associated LED (KNX Programming LED) is indicating programming mode by red lighting.
In case of a KNX bus failure, the LED is blinking. Blinking stops after bus recovery.

Avoid pressing ***"Prog."*** button while start-up of device: If ***"Prog."*** button is pressed while device is starting up, it will not go into operating mode. Instead starting normally, device will only show a rapid blinking of the ***"Pwr/Test"*** LED. Re-starting device without pressing ***"Prog."*** button will put device back into operating mode. Even if not in operating mode, device can still be programmed through KNX-Bus, but ***Handmode*** will be disabled.

## 2.2.2 "Pwr / Test" LED and *Handmode*

In normal operation, the green ***"Pwr/Test"*** LED is continuously ON, indicating the device is powered and running.
In ***Handmode***, the ***"Pwr/Test"*** LED is blinking slowly.

If ***"Pwr/Test"*** LED is blinking rapidly, device has **not** started normally. This may due to:

- ***"Prog."*** button was pressed at start-up of device, so simply re-start device. Refer to 2.2.1
- ***"Test"*** button was pressed at start-up of device, so a master reset occurred: device has to be re-programmed through KNX bus. Refer to 2.2.4
- Memory inside device is inconsistent. If device detects inconsistent memory, it executes a **master reset**, erasing all parameters, physical address and group objects: device has to be re-programmed through KNX bus.

In all this cases, the ***Handmode*** is disabled.

***Handmode:***
In order to test correct installation, the device can be operated manually, without need of KNX connection or programming.
Refer to 2.2.4

## 2.2.3 "Overload" LED

The output current on 24VAC is continuously monitored inside device.
Short or small overloads are admitted. If small overloads are lasting too long, or if even short overloads are too high, the 24VAC supply of the valve is switched OFF for a given time, and will automatically recover.
After recovering, the same measurement takes place again.
So, overheating of device due to overloads on valve due to miss-wiring or short circuit can be reduced significantly.
The ***"Overload"*** LED will be red lighting to indicate the overload condition. Once the overload is removed, the LED stops lighting.
After device restarting, the ***"Overload"*** LED is also lighting for a short time.

## 2.2.4 "Test", increment "↑" and decrement "↓" buttons

Pressing ***"Test"*** button for at least 3 seconds will make device entering ***Handmode***.
Once ***Handmode*** is entered, the green "**Pwr/Test"** LED will start blinking slowly, the fan will switch to speed 0, cooling relay will switch OFF and valve output will be set to 5V if in analog mode (factory setting), or 50% pulse ratio if in PWM mode.

While in ***Handmode***, short presses on ***"Test"*** button will increment cyclically the fan speed.
Once the highest fan speed is reached (factory setting = 3 fan speeds), the next short press will set the fan speed back to 0.
Switching between fan speeds will be done respecting actual "Fan starting time" (factory setting = 2 seconds) and "Fan stopping time" (factory setting = 1 second).

***Handmode*** will exit automatically if no button is pressed for 15 seconds or if pressing the ***"Test"*** button again for at least 3 seconds.

Pressing shortly the ***"↑"*** button while in ***Handmode*** will increment the valve output of about 0.625V if in analog mode or 6.25% pulse ratio if in PWM mode until maximum output is reached (10V if in analog mode (factory setting), or 100% pulse ratio if in PWM mode).

Pressing ***"↑"*** button for at least 3 seconds while in ***Handmode*** will switch the cooling relay ON.

Pressing shortly the **"↓"** button while in ***Handmode*** will decrement the valve output of about 0.625V if in analog mode or 6.25% pulse ratio if in PWM mode until minimum output is reached (0V if in analog mode (factory setting), or 0% pulse ratio if in PWM mode).

Pressing ***"↓"*** button for at least 3 seconds while in ***Handmode*** will switch the cooling relay OFF.

Avoid pressing ***"Test"*** button while start-up of device: If ***"Test"*** button is pressed while device is starting up, it will execute a **master reset**, erasing all parameters, physical address and group objects. Instead starting normally, device will only show a rapid blinking of the ***"Pwr/ Test"*** LED. Even re-starting device will not recover erased data's. ***Handmode*** is also disabled. Device has first to be re-programmed through KNX-Bus before recovering functionality.

# 3. Application description

## *3.1. Communication objects*

Here are listed all communication objects (CO's) of device. According to the settings done within the parameters, some of the CO's may NOT be visible because not relevant for some settings.

| Number | Name | Object Function | Descripti... | Group Address... | Length | ... | R | W | T | U | Data Type | Priority |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Status device | | | | 1 bit | C | - | - | T | - | state | Low |
| 1 | Forcing mode | | | | 1 bit | C | - | W | - | - | enable | Low |
| 2 | Fan automatic mode | | | | 1 bit | C | - | W | - | - | enable | Low |
| 3 | Fan manual mode | | | | 1 bit | C | - | W | - | - | enable | Low |
| 4 | Status automatic mode | | | | 1 bit | C | - | - | T | - | state | Low |
| 5 | Status manual mode | | | | 1 bit | C | - | - | T | - | state | Low |
| 6 | Fan manual speed 0-100% | | | | 1 Byte | C | - | W | - | - | percentage (0..100%) | Low |
| 7 | Fan manual speed counter | | | | 1 Byte | C | - | W | - | - | counter pulses (0..255) | Low |
| 8 | Fan manual speed 1 | | | | 1 bit | C | - | W | - | - | enable | Low |
| 9 | Fan manual speed 2 | | | | 1 bit | C | - | W | - | - | enable | Low |
| 10 | Fan manual speed 3 | | | | 1 bit | C | - | W | - | - | enable | Low |
| 11 | Fan manual speed changing | | | | 1 bit | C | - | W | - | - | step | Low |
| 12 | Status fan speed | | | | 1 Byte | C | - | - | T | - | counter pulses (0..255) | Low |
| 13 | Status fan speed 1 | | | | 1 bit | C | - | - | T | - | state | Low |
| 14 | Status fan speed 2 | | | | 1 bit | C | - | - | T | - | state | Low |
| 15 | Status fan speed 3 | | | | 1 bit | C | - | - | T | - | state | Low |
| 16 | Status temperature display | | | | 2 Byte | C | - | - | T | - | temperature (°C) | Low |
| 17 | Status temperature regulation | | | | 2 Byte | C | - | - | T | - | temperature (°C) | Low |
| 18 | KNX window contact | | | | 1 bit | C | - | W | - | - | open window or door | Low |
| 19 | Status window open mode | | | | 1 bit | C | - | - | T | - | state | Low |
| 20 | Status binary input | | | | 1 bit | C | - | - | T | - | state | Low |
| 21 | Cooling switch | | | | 1 bit | C | - | W | - | - | on/off | Low |
| 22 | Status cooling relay | | | | 1 bit | C | - | - | T | - | state | Low |
| 23 | Cooling 0-100% | | | | 1 Byte | C | - | W | - | - | percentage (0..100%) | Low |
| 24 | Status cooling 0-100% | | | | 1 Byte | C | - | - | T | - | percentage (0..100%) | Low |
| 25 | Status cooling output | | | | 1 Byte | C | - | - | T | - | percentage (0..100%) | Low |
| 26 | Status overload 24VAC | | | | 1 bit | C | - | - | T | - | alarm | Low |
| 27 | Status 24VAC power | | | | 1 Byte | C | - | - | T | - | counter pulses (0..255) | Low |

Figure 4: Communication object list

### 3.1.1 *Status device*

The object 0 is for the device status.

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 0 | *Status device* | Inactive / Active | 1 bit | C/T |
| 1: Device is running<br>0: -<br>The device will only send 1. This CO can be used for monitoring the device if set to be sent cyclically, so failing devices can be detected. | | | | |

**(Red values are default)**

### 3.1.2 *Forcing mode*

The object 1 is for forcing the device.

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 1 | ***Forcing mode*** | Disable / Enable | 1 bit | C/W |
| 0: Set device to exit ***forcing mode***<br>1: Set device to work in ***forcing mode***<br>Sending a "1" on CO Nr. 1/ will place the device in ***forcing mode***. In ***forcing mode***, fan, cooling relay and valve are forced to levels set by parameter.<br>***Forcing mode*** has highest priority compared to other modes (***manual mode***, ***automatic mode*** or ***window open mode***).<br>Only ***Handmode*** has higher priority. | | | | |

### 3.1.3 *Fan automatic mode*

The object 2 is for the fan ***automatic mode***

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 2 | *Fan* ***automatic mode*** | Disable / Enable | 1 bit | C/W |
| 0: Set fan to work in ***manual mode*** and re-starts fall back time in ***automatic mode***<br>1: Set fan to work in ***automatic mode***<br>Sending a "1" on CO Nr. 2/ will place the device in ***automatic mode***.<br>In ***automatic mode***, fan speeds are driven automatically, depending on communication Object Nr. 23/, *Cooling 0-100%.*<br>Sending a "0" on CO Nr. 2/ will put device in ***manual mode*** and re-starts fall back time in ***automatic mode***.<br>Also refer to CO Nr. 3/.<br>CO's Nr. 2 and Nr. 3 are complementary. | | | | |

The ***automatic mode*** is always set default at restart the device.

### 3.1.4 *Fan manual mode*

The object 3 is for the fan ***manual mode***

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 3 | *Fan* ***manual mode*** | Disable / Enable | 1 bit | C/W |
| 0: Set fan to work in ***automatic mode***<br>1: Set fan to work in ***manual mode*** and re-starts fall back time in ***automatic mode***<br>Sending a "1" on CO Nr.3/ will place the device in ***manual mode***.<br>In ***manual mode***, the fan speeds are driven manually, using the CO's Nr. 6 – 11.<br>After a given time, device will fall back in ***automatic mode***. This time can be set in parameters. The fall back can also be de-activated.<br>The duration of this time is also re-started after each action of CO's Nr. 6 – 11 or by putting device in ***manual mode*** by sending "0" to CO Nr. 2/.<br>Sending a "0" on CO Nr. 3/ will put device in ***automatic mode.***<br>Also refer to CO Nr. 2/.<br>CO's Nr. 2 and Nr. 3 are complementary. | | | | |

### 3.1.5 *Status automatic mode*

The object 4 is for the ***automatic mode*** status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 4 | *Status* ***automatic mode*** | Inactive / Active | 1 bit | C/T |
| 0: Fan is in not in ***automatic mode*** (can be in ***manual mode*** or in ***forcing mode***, see fig. 5)<br>1: Fan is in ***automatic mode***<br>CO Nr. 4/ indicates if device is in ***automatic mode***.<br>Refer to CO Nr. 2/, Nr. 3/ and Nr. 5/. | | | | |

### 3.1.6 *Status manual mode*

The object 5 is for the ***manual mode*** status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 5 | *Status* ***manual mode*** | Inactive / Active | 1 bit | C/T |
| 0: Fan is not in ***manual mode*** (can be in ***automatic mode*** or in ***forcing mode***, see fig. 5)<br>1: Fan is in ***manual mode***<br>CO Nr. 5/ indicates if device is in ***manual mode***<br>Refer to CO Nr. 2/, Nr. 3/ and Nr. 4/. | | | | |

| | | CO Nr. 4/, *Status* ***automatic mode*** | |
|---|---|---|---|
| | | 0 | 1 |
| CO Nr. 5/, *Status* ***manual mode*** | 0 | ***Forcing mode*** | ***Automatic mode*** |
| | 1 | ***Manual mode*** | Does not apply |

Figure 5: device working mode

### 3.1.7 *Fan manual speed 0-100%*

The object 6 is for the setting a manual fan speed using a 0-100% value

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 6 | *Fan manual speed 0-100%* | Percentage[0..100%] | 1 Byte | C/W |

0: Switch to 0% fan speed
….
255: Switch to 100% fan speed
Sending a value on CO Nr. 6/ will switch the fan speed according to the individual thresholds set in parameters, and put the device in ***manual mode***. Refer to CO Nr. 3/.
This will also re-start the fall back time in ***automatic mode***.

### 3.1.8 *Fan manual speed counter*

The object 7 is for the setting a manual fan speed using counter pulses

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 7 | *Fan manual speed counter* | Counter pulses[0..255] | 1 Byte | C/W |

0: Switch fan speed 0
1: Switch fan speed 1
2: Switch fan speed 2
3: Switch fan speed 3
4 … 255: Switch maximum fan speed
Sending a value on CO Nr. 7/ will switch the according fan speed and put the device in ***manual mode***. Refer to CO Nr. 3/.
This will also re-start the fall back time in ***automatic mode***.
If sending a number greater than the numbers of fan speeds set in parameters, the maximum fan speed will be set.

### 3.1.9 *Fan manual speed 1*

The object 8 is for switching fan to manual speed 1

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 8 | *Fan manual speed 1* | Disable / Enable | 1 bit | C/W |

0: Switch fan OFF
1: Switch fan speed 1
Sending a “1” on CO Nr. 8/ will switch to fan speed 1 and put the device in ***manual mode***.
Refer to CO Nr. 3/
Sending a “0” on CO Nr. 8/ will switch the fan OFF and put the device in ***manual mode***.
Sending to this CO also re-start fall back time in ***automatic mode***.

### 3.1.10 *Fan manual speed 2*

The object 9 is for switching fan to manual speed 2

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 9 | *Fan manual speed 2* | Disable / Enable | 1 bit | C/W |
| 0: Switch fan OFF<br>1: Switch fan speed 2<br>Sending a "1" on CO Nr. 9/ will switch to fan speed 2 and put the device in ***manual mode***.<br>Refer to CO Nr. 3/<br>Sending a "0" on CO Nr. 9/ will switch the fan OFF and put the device in ***manual mode***.<br>Sending to this CO also re-start fall back time in ***automatic mode***. | | | | |

### 3.1.11 *Fan manual speed 3*

The object 10 is for switching fan to manual speed 3

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 10 | *Fan manual speed 3* | Disable / Enable | 1 bit | C/W |
| 0: Switch fan OFF<br>1: Switch fan Speed 3<br>Sending a "1" on CO Nr. 10/ will switch to fan speed 3 and put the device in ***manual mode***.<br>Refer to CO Nr. 3/<br>Sending a "0" on CO Nr. 10/ will switch the fan OFF and put the device in ***manual mode***.<br>Sending to this CO also re-start fall back time in ***automatic mode***. | | | | |

### 3.1.12 *Fan manual speed changing*

The object 11 is for manual fan speed changing (stepping)

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 11 | *Fan manual speed changing* | Decrease / Increase | 1 bit | C/W |
| 0: decrease fan speed 3 -> 2 -> 1 -> 0<br>1: increase fan speed 0 -> 1 -> 2 -> 3 (limited to maximum number of fan speeds set in parameters)<br>Sending a "1" on CO Nr. 11/ will increment the fan speed and put the device in ***manual mode***.<br>Incrementing will be limited to the maximum number of fan speeds set in parameters.<br>Sending a "0" on CO Nr. 11/ will decrement the fan speed and put the device in ***manual mode***.<br>Refer to CO Nr. 3/.<br>Sending to this CO also re-start fall back time in ***automatic mode***. | | | | |

### 3.1.13 *Status fan speed*

The object 12 is for fan speed status.

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 12 | *Status fan speed* | Counter pulses[0..255] | 1 Byte | C/T |

0: Fan is OFF
1: Fan is in speed 1
2: Fan is in speed 2
3: Fan is in speed 3
CO Nr. 12/ indicates the actual fan speed.
Note: The OFF-time between different fan speeds is NOT displayed

### 3.1.14 *Status fan speed 1*

The object 13 is for fan speed 1 status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 13 | *Status fan speed 1* | Inactive / Active | 1 bit | C/T |

0: Fan is NOT in speed 1
1: Fan is in speed 1
CO Nr. 13/ indicates if the actual fan speed is 1
Note: during the OFF-time after changing from speed 1, status speed 1 keeps STILL active.

### 3.1.15 *Status fan speed 2*

The object 14 is for fan speed 2 status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 14 | *Status fan speed 2* | Inactive / Active | 1 bit | C/T |

0: Fan is NOT in speed 2
1: Fan is in speed 2
CO Nr. 14/ indicates if the actual fan speed is 2
Note: during the OFF-time after changing from speed 2, status speed 2 keeps STILL active.

### 3.1.16 *Status fan speed 3*

The object 15 is for fan speed 3 status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 15 | *Status fan speed 3* | Inactive / Active | 1 bit | C/T |

0: Fan is NOT in speed 3
1: Fan is in speed 3
CO Nr. 15/ indicates if the actual fan speed is 3
Note: during the OFF-time after changing from speed 3, status speed 3 keeps STILL active.

### 3.1.17 *Status temperature display*

The object 16 is for temperature display status (from local temperature sensor)

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 16 | *Status temperature display* | Temperature (°C) | 2 Byte | C/T |

Encoding: MEEEEMMM MMMMMMMM
Float value = (0.01*M)*2^(E)
E = [0 … 15]
M = [-2048 … 2047], 2's complement notation
7FFFh denotes invalid data
CO Nr. 16/ indicates the temperature measured from local sensor in steps of 0.5°C.
This temperature is calculated with a hysteresis of 0.5°C to avoid jittering, and so is best suited for use on a temperature display
If no temperature compensation is used, the device will send values between 0.0°C and 50.0°C.

### 3.1.18 *Status temperature regulation*

The object 17 is for temperature regulation status (from local temperature sensor)

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 17 | *Status temperature regulation* | Temperature (°C) | 2 Byte | C/T |

Encoding: MEEEEMMM MMMMMMMM
Float Value = (0.01*M)*2^E
E = [0 … 15]
M = [-2048 … 2047], 2's complement notation
7FFFh denotes invalid data
CO Nr. 17/ indicates the temperature measured from local sensor in steps of 0.1°C.
This temperature is in steps of 0.1°C, without hysteresis for best accuracy, and so is best suited for use for regulation
Because of its fast reaction it is not intended to be used for display
If no temperature compensation is used, the device will send values between 0.0°C and 50.0°C.

### 3.1.19 *KNX window contact*

The object 18 is for KNX window contact

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 18 | *KNX window contact* | Closed / Open | 1 bit | C/W |
| 0: Window is closed<br>1: Window is open<br>If using a KNX window contact, his state can be sent to device by using CO Nr. 18/<br>Sending a "1" on CO Nr. 18/ indicate the device that the window on KNX contact is open. | | | | |

By default, prior to receiving CO Nr.18/, *KNX window contact,* window is considered as closed.

### 3.1.20 *Status window open mode*

The object 19 is for ***window open mode*** status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 19 | *Status* ***window open mode*** | Inactive / Active | 1 bit | C/T |
| 0: Device is NOT working in ***window open mode***<br>1: Device is working in ***window open mode***<br>CO Nr. 19/ indicates if the device is actually working in ***window open mode***.<br>Depending on parameters, the ***window open mode*** is started / ended according to external KNX window contact on CO Nr. 18/, local window contact and the delay windows are actually open or closed. | | | | |

### 3.1.21 *Status binary input*

The object 20 is for binary input status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 20 | *Status binary input* | Inactive / Active | 1 bit | C/T |
| 0: Binary input is inactive<br>1: Binary input is active<br>CO Nr. 20/ indicates the status of the local binary input. The polarity of the binary input can be set by parameter. | | | | |

## 3.1.22 *Cooling switch*

The object 21 is for driving the cooling relay / enabling the cooling valve

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 21 | *Cooling switch* | Off / On | 1 bit | C/W |

0: Switches OFF cooling relay / Disables 0-10VDC or PWM output
1: Switches ON cooling relay / Enables 0-10VDC or PWM output
According to parameters, CO Nr. 21/ can be used either to enable / disable the cooling valve output, and / or to switch ON / OFF the cooling relay.
Sending "1" on CO Nr. 21/ enables the valve output / switches ON the cooling relay.
Sending "0" on CO Nr. 21/ disables the valve output / switches OFF the cooling relay.
With parameters, this object can be disabled, and cooling relay will switch according to threshold values of CO Nr. 23/ *Cooling 0-100%.*

## 3.1.23 *Status cooling relay*

The object 22 is for cooling relay status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 22 | *Status cooling relay* | Inactive / Active | 1 bit | C/T |

0: Cooling relay is OFF
1: Cooling relay is ON
CO Nr. 22/ indicates the actual state of the relay.

## 3.1.24 *Cooling 0-100%*

The object 23 is for Cooling 0-100%

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 23 | *Cooling 0-100%* | Percentage [0..100%] | 1 Byte | C/W |

0: 0% Cooling
….
255: 100% Cooling
CO Nr. 23/ is usually connected to the regulating output of a room temperature controller.
Depending on settings, sending a value on CO Nr. 23 can drive the cooling valve, set the fan speed and the cooling relay.

By default, prior to receiving CO Nr. 23/, *cooling 0-100%,* is considered 0%.

## 3.1.25 *Status cooling 0-100%*

The object 24 is for cooling 0-100% status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 24 | *Status cooling 0-100%* | Percentage [0..100%] | 1 Byte | C/T |

0: 0% Cooling
…
255: 100% Cooling
CO Nr. 24/ returns the actual relative position of the cooling valve between closed valve and full open valve.
It gives an indication of the cooling ratio, taking also into account ***open window mode***, ***forcing mode***, default, …
So it is well better suited for display on a room temperature controller as CO Nr. 23/.

## 3.1.26 *Status cooling output*

The object 25 is for cooling output status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 25 | Status cooling output | Percentage [0..100%] | 1 Byte | C/T |

0: 0% = 0V cooling / 0% pulse ratio
…
255: 100% = 10V cooling / 100% pulse ratio
CO Nr. 25/ returns the actual voltage (if analog) or pulse ratio (if PWM) of the cooling valve, taking also into account ***open window mode***, ***forcing mode***, default, …
So it is best suited for maintenance or checking purpose.

## 3.1.27 *Status overload 24VAC*

The object 26 is for overload 24VAC status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 26 | *Status overload 24VAC* | No alarm / Alarm | 1 bit | C/T |

0: 24VAC is OK
1: 24VAC is overloaded
CO Nr. 26/ indicates if an overload on the cooling valve occurred.
The current feeding the cooling valve is continuously monitored.
Short or small overloads are admitted. If small overloads are lasting too long, or if even short overloads are too high, the 24VAC supply of the valve is switched OFF for a given time, and will automatically recover.
After recovering, the same measurement takes place again.
So, overheating of device due to overloads on valve due to miss-wiring or short circuit can be reduced significantly.
The overload status is filtered to avoid jittering. It is also indicated on device by a red LED.

### 3.1.28 *Status 24VAC power*

The object 27 is for 24VAC power status

| Number | Name | Function | Length | Flags |
|---|---|---|---|---|
| 27 | *Status 24VAC power* | Counter pulses[0..255] | 1 Byte | C/T |

0: 0VA cooling
1: 0.25VA cooling
…
18: 4.5VA cooling (maximum nominal power)
…
255:
CO Nr. 27/ indicates an approximated power consumption of the cooling valve. This is principally used internally for overload protection, but may be of interest for installer to check correct valve wiring and function.
A valve without any consumption when it should move may be defective or wires may be unconnected.
A valve with too much consumption may be defective or wiring may be short circuited.

## *3.2 Parameter description*

### 3.2.1 General parameters

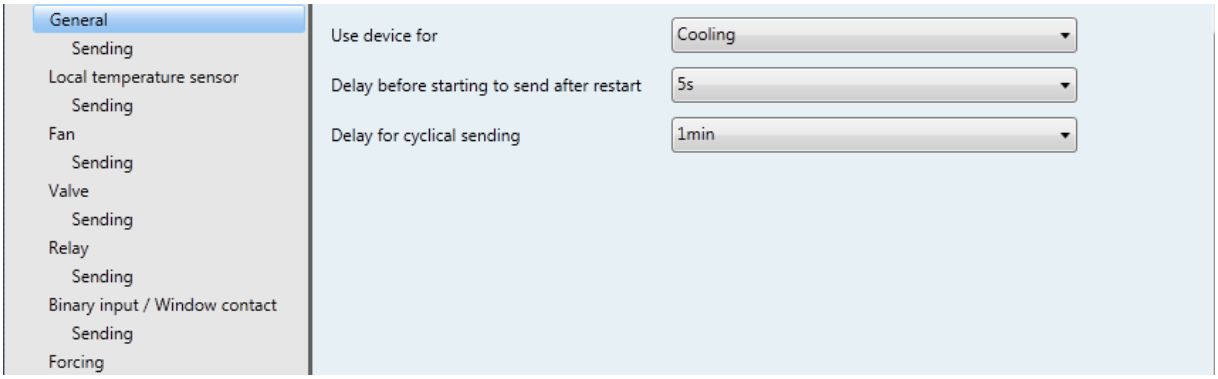



Figure 6: General parameters

**Use device for**
This parameter is used to indicate if the device will be used for heating or for cooling. The main incidence will be the displayed name of the communication objects and of parameters.

**Delay before starting to send after restart**
In order to avoid bus overload after restart or recovery of a complete system, the *delay before starting to send after restart* can be set individually for each device.
So, the device will not send out any CO's before this delay is elapsed.
This delay is applicable after power return, programming of device or recovery from bus failure.
The delay set in general parameters is concerning all status CO's of device.
The processor also need about 3 seconds to start.

**Delay for cyclical sending**
All CO's set by parameters to be sent out cyclically will be sent out together in given time delays. This delay can be set with this parameter. The cyclical sending will only begin after the "Delay before starting to send after restart" is elapsed.
Choosing "No cyclical sending" will deactivate cyclical sending of all CO's.
The delay set in "general parameters" is concerning all status CO's of device.

## 3.2.2 Local temperature sensor parameters

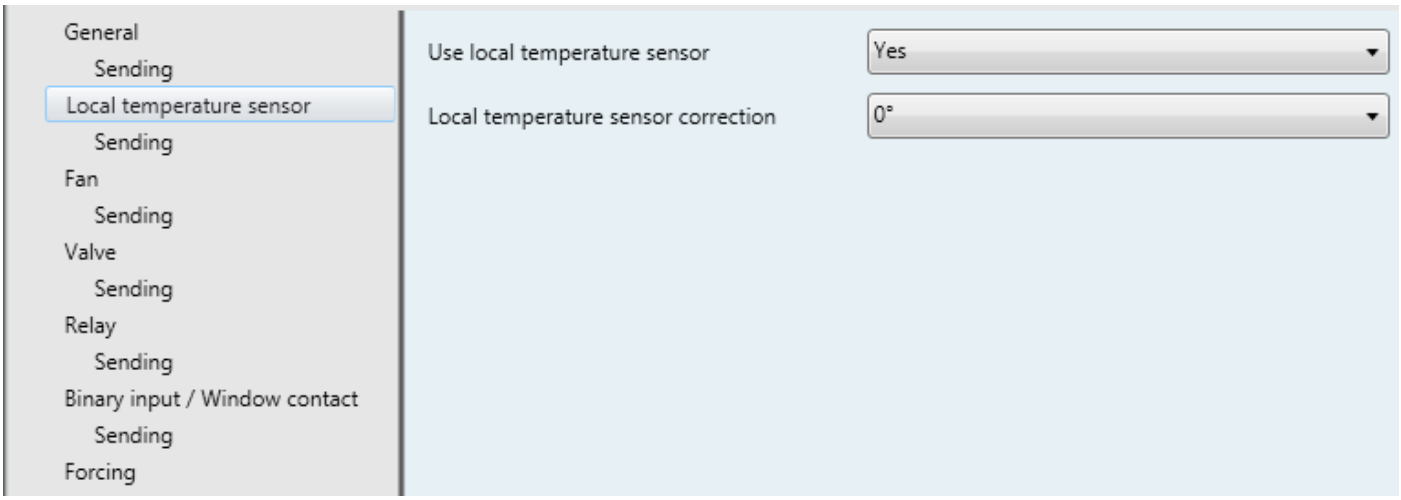



Figure 7: Local temperature sensor parameters

A local temperature sensor may be connected to the device. The measured temperature is NOT used inside device, but only to be sent out on bus, e.g. to be used by a room temperature controller or for a display.

### Use local temperature sensor

This parameter is used to indicate if a local temperature sensor is connected to device or not.

### Local temperature sensor correction

In order to compensate incorrect temperature measurements due to not optimal placement, the measured temperature can be adjusted by adding or subtracting a fixed value.

| | Min. | Max. |
|---|---|---|
| **Measured temperature range** | 0.0°C | +50.0°C |
| **Temperature adjustment range** | -5.0°C | +5.0°C |
| **Corrected temperature range** | 0.0°C | +55.0°C |

For all values outside of these ranges, value 0x7FFFh, denoting invalid data, will be used.

Figure 8a: Adjusted temperature vs. measured temperature

The corrected temperature can be sent out on bus using 2 different CO's. 16 and 17.
For more detailed information about the single status CO's refer to the communication object description.

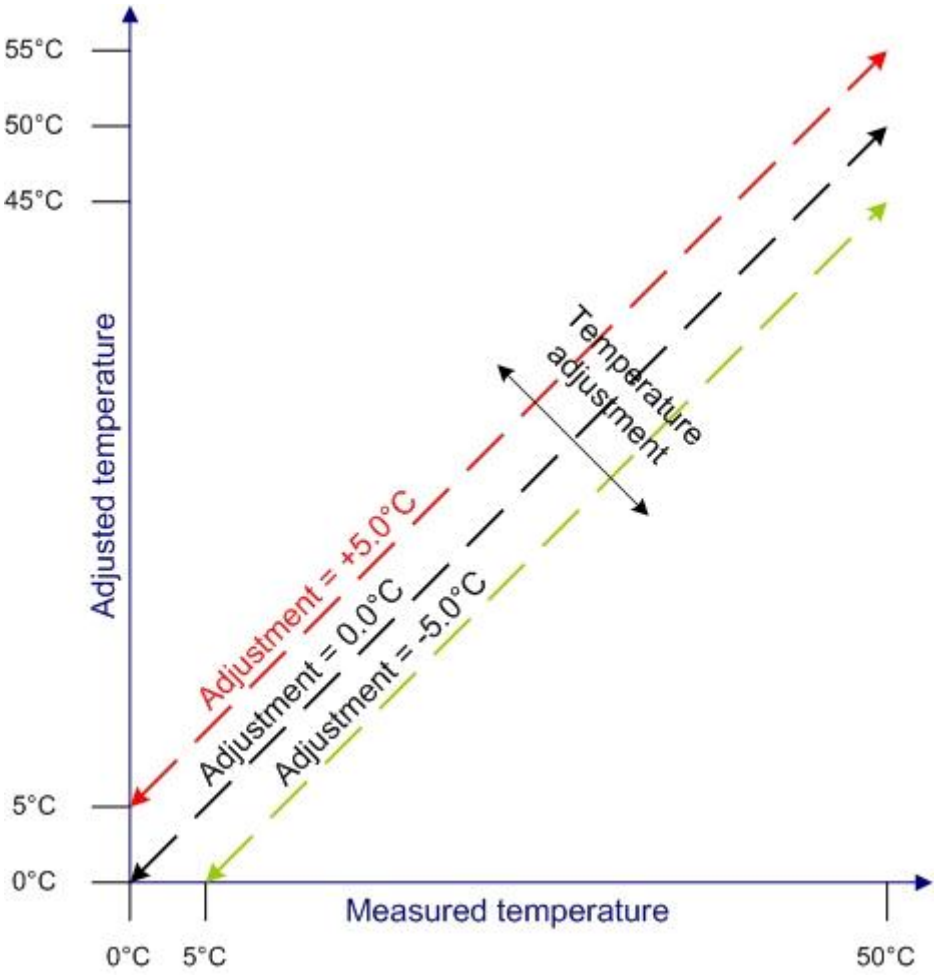



Figure 8b: Adjusted temperature vs. measured temperature

## 3.2.3 Fan parameters

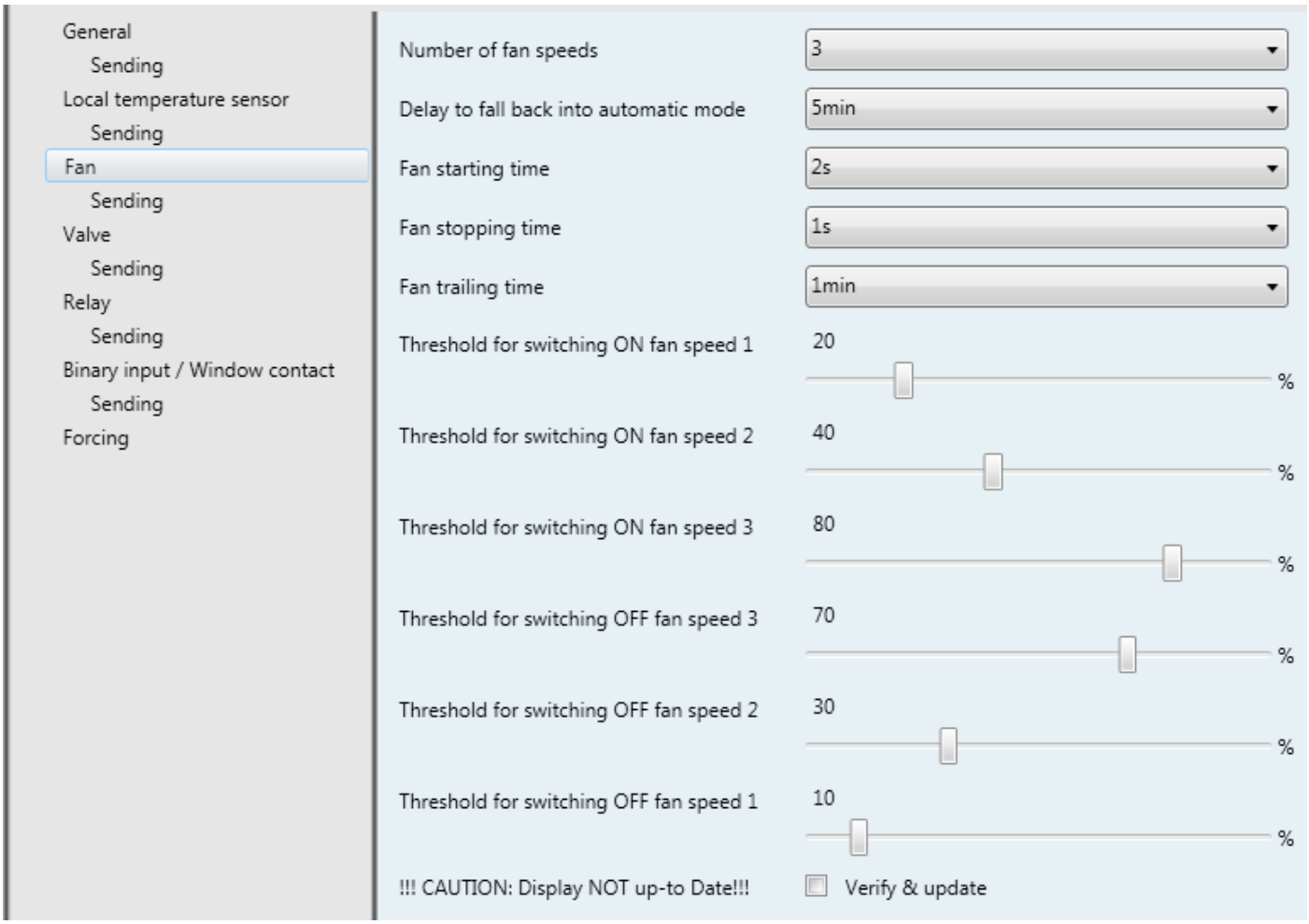



Figure 9: Fan parameters

The device can be used to drive fans with up to 3 speeds. Only 1 speed at a time can be activated.

### Number of fan speeds

This parameter is used to indicate how many fan speeds are to be used.

### Delay to fall back into automatic mode

Usually and to achieve best regulation and efficiency, the device should work in ***automatic mode***. This is the normal operation, where the actual fan speed is calculated depending on communication object Nr. 23/, *Cooling 0-100%.*
According to this value, usually calculated by a room temperature controller, the device will switch the appropriate fan speed.

Sometimes, for comfort reasons or for maintenance, the device will be put into ***manual mode***.
The return to ***automatic mode*** can be initiated by some CO's, but also by internal delayed fall back.
The delay to wait before falling back in ***automatic mode*** can be set with this parameter.
Each action of ***manual mode*** (refer to communication object description) will re-start the delay counter.
This parameter also permits to de-activate the fall back in ***automatic mode***, but this feature should be used with caution: in ***manual mode***, the temperature regulation will not be efficient. For example, even if the cooling valve is fully open, room overheating could not be avoided if fan is still manually blocked in speed 0.

**Fan starting time**
This parameter is used to set the minimum time a fan speed can be switched ON.
Until this time isn't elapsed, the device will not switch OFF the actual fan speed.
This will prevent the relays to switch OFF while the fan is still starting and absorbing higher currents.
So, by preventing to switch relays OFF during fan inrush currents, the electrical lifetime of device can be expanded.
Furthermore, permanent and annoying switching between different fan speeds can be reduced.
Even ***forcing mode*** will NOT overrun this delay.

**Fan stopping time**
This parameter is used to set the minimum OFF time between fan speeds.
Until this time isn't elapsed, the device will not switch ON any fan speed.
This will prevent the relays to switch ON while the fan is still rotating.
So, by preventing to switch relays ON during fan is still rotating, the electrical lifetime of device can be expanded.
Furthermore, permanent and annoying switching between different fan speeds can be reduced.
Even ***forcing mode*** will NOT overrun this delay.

**Fan trailing time**
In ***automatic mode***, this parameter is used to set the time the fan should continue in speed 1 even after it should normally switch OFF.
In many applications, even if the cooling valve closes, the cooling unit still keeps cold for a long time if the fan is switched OFF. Thus, some condensation may result and create moisture. Still keeping the fan ON in speed 1 will temperate the cooling unit, and so help reducing condensation and moisture.

**Threshold for switching ON fan speed 1**
In ***automatic mode***, if value of CO Nr. 23/, *Cooling 0-100%* or CO Nr.6/, *Fan manual speed 0-100%* is higher than value set for this parameter, at least fan speed 1 will be switched ON.
This avoids starting the fan for small regulating outputs from room temperature controller, and so decreasing disturbing noise.

**Threshold for switching ON fan speed 2**
In ***automatic mode***, if value of CO Nr. 23/, *Cooling 0-100%* or CO Nr.6/, *Fan manual speed 0-100%* is higher than value set for this parameter, at least fan speed 2 will be switched ON.

**Threshold for switching ON fan speed 3**
In ***automatic mode***, if value of CO Nr. 23/, *Cooling 0-100%* or CO Nr.6/, *Fan manual speed 0-100%* is higher than value set for this parameter, fan speed 3 will be switched ON.

**Threshold for switching OFF fan speed 3**
In ***automatic mode***, if value of CO Nr. 23/, *Cooling 0-100%* or CO Nr.6/, *Fan manual speed 0-100%* is lower than value set for this parameter, fan speed 3 will be switched OFF.

**Threshold for switching OFF fan speed 2**
In ***automatic mode***, if value of CO Nr. 23/, *Cooling 0-100%* or CO Nr.6/, *Fan manual speed 0-100%* is lower than value set for this parameter, fan speed 2 will be switched OFF.

**Threshold for switching OFF fan speed 1**
In ***automatic mode***, if value of CO Nr. 23/, *Cooling 0-100%* or CO Nr.6/, *Fan manual speed 0-100%* is lower than value set for this parameter, fan will be switched OFF, once fan trailing time is elapsed.

**The thresholds to switch between the different fan speeds can be set individually, but following rules have to be respected:**

Threshold for switching ON fan speed 1 ≤ Threshold for switching ON fan speed 2 ≤ Threshold for switching ON fan speed 3

Threshold for switching OFF fan speed 1 ≤ Threshold for switching OFF fan speed 2 ≤ Threshold for switching OFF fan speed 3

Threshold for switching OFF fan speed 1 ≤ Threshold for switching ON fan speed 1
The difference should be great enough to avoid frequent and disturbing fan speed changing.

Threshold for switching OFF fan speed 2 ≤ Threshold for switching ON fan speed 2
The difference should be great enough to avoid frequent and disturbing fan speed changing.

Threshold for switching OFF fan speed 3 ≤ Threshold for switching ON fan speed 3
The difference should be great enough to avoid frequent and disturbing fan speed changing.

**Verify & Update**

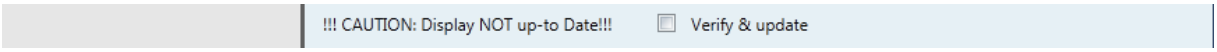



Figure 10: Verify & update

This check box appears once new values have been set.
The input values are automatically checked and adjusted to respect the above rules, but only clicking this box will update the displayed values. So, this box is never to be left unchecked.

For more detailed information about the single status CO's refer to the communication object description.

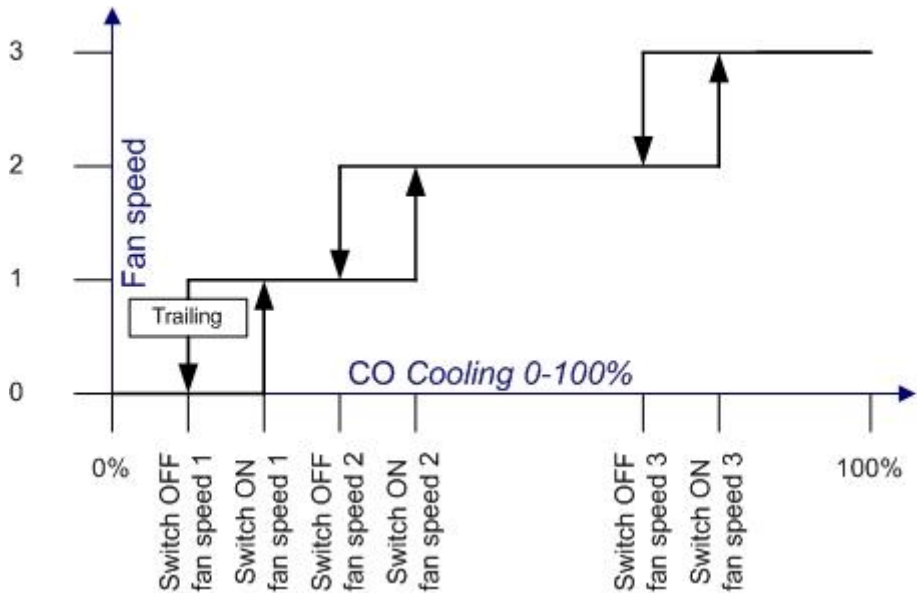



Figure 11: Fan speed vs. CO *Cooling 0-100%* in ***automatic mode***

## 3.2.4 Valve parameters

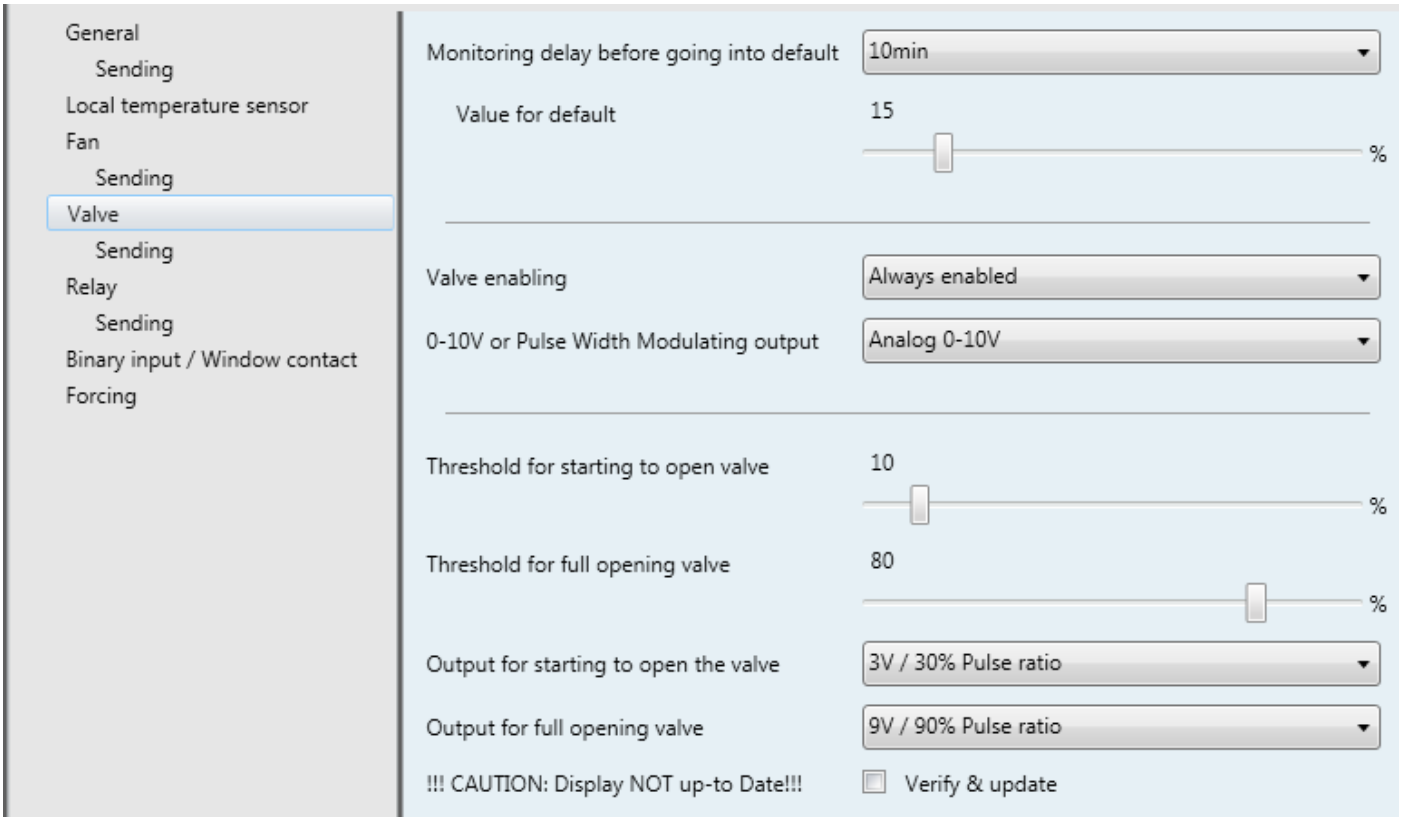



Figure 12: Valve parameters

The device can be used to drive a valve for regulating cooling or heating flow, depending on application.
This valve is normally driven by the CO Nr. 23/, *Cooling 0-100%.*

### Monitoring delay before going into default

In normal operation, the valve is driven by the CO Nr. 23/, *Cooling 0-100%,* usually calculated by a room temperature controller. To avoid overheating or frosting due to bus failure or missing room temperature controller, a monitoring delay can be set with this parameter.
If no CO Nr. 23/, *Cooling 0-100%*, has not been received within the given delay, a default value can be used.
This may also have incidence on fan speed (in ***automatic mode***) and cooling relay (depending on parameters).

### Value for default

This parameter is used to set the default value to be used if a monitoring failure occurs.

### Valve enabling

This parameter is used to indicate how to enable the valve.
If “Always enabled” is selected, CO Nr. 21/, *Cooling switch*, has no incidence on the valve output.
If “By communication object cooling switch” is selected, the valve will only be enabled once a “1” has been received on CO Nr. 21/, *Cooling switch*.
Not receiving this “1” or receiving a “0” will disable the valve.
In normal operation, the enabled valve will be driven by CO Nr. 23/, *Cooling 0-100%*, usually calculated by a room temperature controller.
The output value for disabled valve can be set by parameter.

### 0-10V or pulse width modulating (PWM) output

This parameter is used to indicate the type of valve used in system.
If choosing "analog 0-10V", a 0-10V modulating signal will be driven on the valve connector, and 24VAC power enabled for feeding valve.
Analog 0-10V valves are usually more accurate than PWM valves.
If choosing "Pulse Width Modulating", the 24VAC power feeding will be modulated by pulse width, changing ratio between 0% (always OFF) and 100% (always ON).
For PWM valves, a supplementary parameter is used to specify the PWM period.
PWM valves are usually less expensive than analog 0-10V valves, but also less accurate.

### PWM period

This parameter is used to set the PWM period if using PWM valves.
The period is the delay between consecutive pulses.
The period should be adapted to the valve.

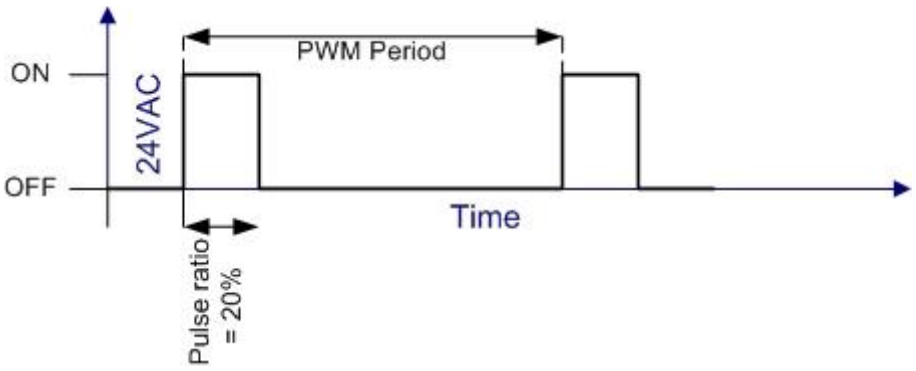



Figure 13: Pulse width modulation: 24VAC vs. time

### Threshold for starting to open valve

This parameter indicates the threshold of CO Nr. 23/, *Cooling 0-100%*, to start to open the valve.
In normal operation, if value of CO Nr. 23/, *Cooling 0-100%* is lower than value set for this parameter, the valve will keep closed (0V if analog valve, 0% ratio if PWM valve).
This avoids having continuously refrigerating or heating flow even for small regulating outputs from room temperature controller, and so decreasing energy consumption.

### Threshold for full opening valve

This parameter indicates the threshold of CO Nr. 23/, *Cooling 0-100%,* to full open the valve.
In normal operation, if value of CO Nr. 23/, *Cooling 0-100%* is higher than value set for this parameter, the valve will keep full open.
This allows full refrigerating or heating flow, without needing maximum regulating output from room temperature controller, and so increasing reactivity of the system.

### Output for starting to open the valve

This parameter is used to set the output value for starting to open the valve. The voltage (if analog 0-10V valve) or pulse ratio (if PWM valve) stored into this parameter will be the minimum value on output if valve has not to be closed. So, it is possible to start with a significant flow as soon as regulating has to take place, increasing reactivity of the system.

### Output for full opening valve

This parameter is used to set the output value for full opening valve. The voltage (if analog valve 0-10V) or pulse ratio (if PWM valve) stored into this parameter will be the maximum value on output if valve has to be opened. So, it is possible to limit maximum flow.

### Output for valve disabled

This parameter appears only if "By communication object cooling switch " for parameter "Valve enabling" is selected.
This parameter is used to set the output value for disabled valve. The voltage (if analog valve 0-10V) or pulse ratio (if PWM valve) stored into this parameter will be driven on output if valve is disabled.

**The valve thresholds and output values can be set individually, but following rules have to be respected:**

Threshold for starting to open valve ≤ Threshold for full opening valve

Output for starting to open the valve ≤ Output for full opening valve

**Verify & update**

This check box appears once new values have been set.
The input values are automatically checked and adjusted to respect the above rules, but only clicking this box will update the displayed values. So, this box is never to be left unchecked.

For more detailed information about the single status CO's refer to the communication object description.

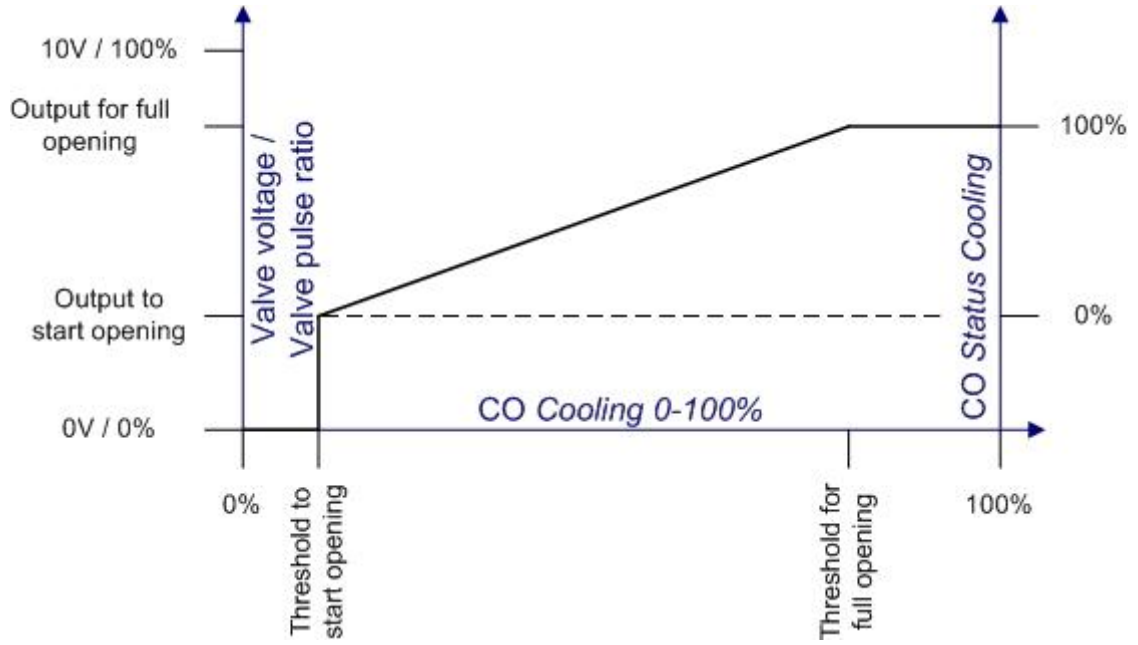



Figure 14: Valve output vs. CO *Cooling 0-100%,* CO *Status Cooling* vs. valve output

## 3.2.5 Relay parameters

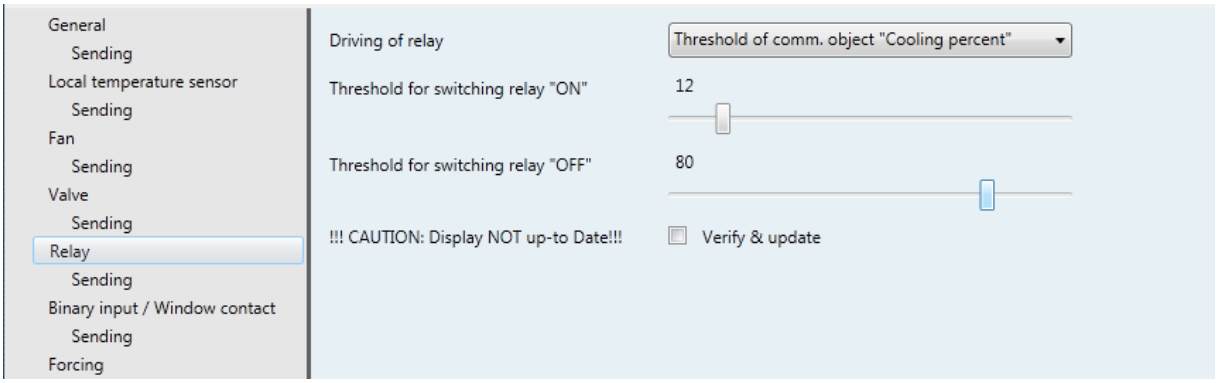



Figure 15: Relay parameters

The device has a supplementary switching relay output. This output can be used for different applications. In combination with valve enabling, it can e.g. be used for switching a circulating pump, or enabling a refrigerant unit.

**Driving of relay**
This parameter is used to indicate how the output relay must be driven.
If choosing "Driven by comm. Object Cooling switch", it will act as a simple switching output, switching ON if receiving a "1" on CO Nr. 21/ *Cooling switch*, and switching OFF if receiving a "0" on CO Nr. 21/ *Cooling switch*.
If choosing "Threshold of comm. Object Cooling percent", the relay will be switched according CO Nr. 23/ *Cooling 0-100%*.

**Threshold for switching relay "ON"**
This parameter is used to set the threshold of CO Nr. 23/ *Cooling 0-100%*, to switch the relay ON.
In normal operation, if value of CO Nr. 23/, *Cooling 0-100%* is lower than value set for this parameter, the relay will be switched OFF.

**Threshold for switching relay "OFF"**
This parameter is used to set the threshold of CO Nr. 23/ *Cooling 0-100%*, to switch the relay OFF.
In normal operation, if value of CO Nr. 23/, *Cooling 0-100%* is higher than value set for this parameter, the relay will be switched ON.

**The relay thresholds values can be set individually, but following rules have to be respected:**

Threshold for switching relay OFF ≤ Threshold for switching relay ON

The difference should be great enough to avoid frequent and disturbing switching.

**Verify & update**
This check box appears once new values have been set.
The input values are automatically checked and adjusted to respect the above rules, but only clicking this box will update the displayed values. So, this box is never to be left unchecked.

For more detailed information about the single Status CO's refer to the communication object description.

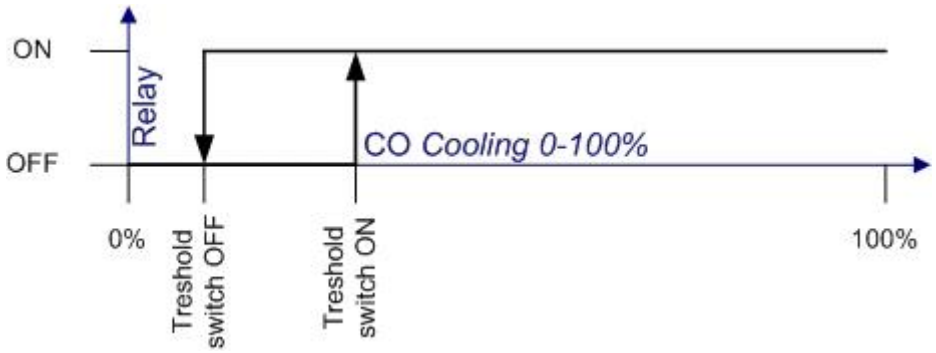



Figure 16: Relay state vs. CO *Cooling 0-100%*

## 3.2.6 Binary input / Window contact parameters

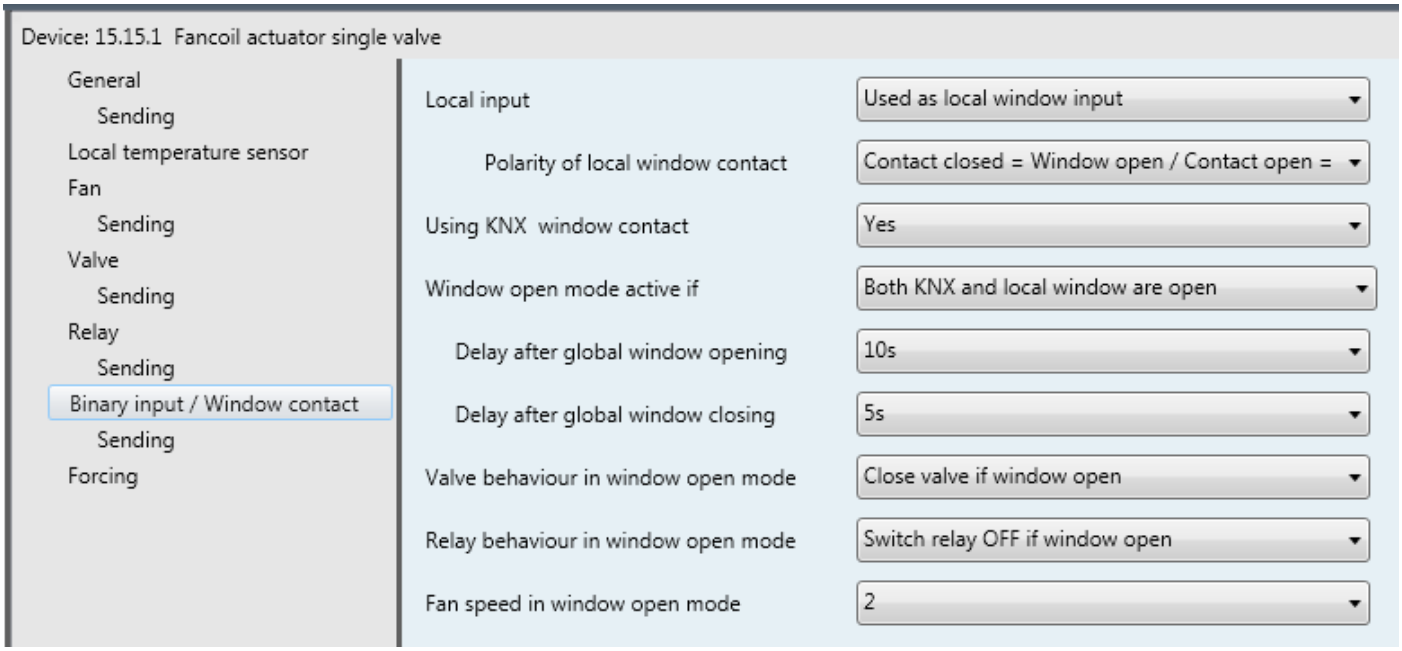



Figure 17: Binary input / Window contact parameters

The device has a digital signal input that can be used either as a general binary input or as a local window contact input.
There is also a communication object for using KNX window contact.
So, there are up to 2 different sources for window contact.
The combination of the local window contact and KNX window contact, according to parameters, will be used to enter a ***window open mode***.
The status of the ***window open mode*** is also available on CO Nr.19/, *Status* ***window open mode***.

The ***window open mode*** can be used to set the device in a predefined state, for example to switch OFF the fan and close the valve if window is open, in order not to waste energy.

### Local input

This parameter is used to indicate how the digital signal input is to be used.
If choosing "Used as binary input", the input signal will simply be transmitted on the bus, without any incidence on the device.
If choosing "Used as local window contact", the digital input will be used to calculate the global window contact.
If choosing "Not used", the digital input will simply be ignored.

### Polarity of binary input

If using a local input, the polarity of the binary input can be set.
If choosing "0 = contact open / 1 = contact closed", the device will send out status "1" on bus if the contact is closed, and send out status "0" if contact is open.
If choosing "0 = contact closed / 1 = contact open", the device will send out status "0" on bus if the contact is closed, and send out status "1" if contact is open.
The status of the binary input is available on CO Nr. 20/, *Status binary input*.

### Polarity of local window contact

If using a local window contact, the polarity of the local window contact can be set.
If choosing "Contact closed = Window open / Contact open = Window closed", the local window will be considered as open if the contact on the binary input is closed.
If choosing "Contact open = Window open / Contact closed = Window closed", the local window will be considered as open if the contact on the binary input is open.

### Using KNX window contact

This parameter is used to indicate if a KNX window contact is to be used.
If choosing "Yes", the signal from an external KNX window contact can be used for calculation of global window contact.
The KNX window contact must be associated to CO Nr. 18/, *KNX Window contact*.

### Window open mode active if

This parameter appears only if both, local window contact and KNX window contact are used.
If choosing "Both KNX and local window are open", both contacts must be signalling open window in order to consider a global window open.
If choosing "At least KNX or local window is open", one single window contact signalling open window will be sufficient to consider a global window open.

### Delay after global window opening

Once global window is calculated as open, it may be necessary to wait a certain delay before entering ***window open mode***. This may avoid the device switching between different modes because of short openings of window.
The status of the ***window open mode*** is available on CO Nr. 19/, *Status* ***window open mode***.

### Delay after global window closing

Once global window is calculated as closed, it may be necessary to wait a certain delay before exiting ***window open mode***. This may avoid the device switching between different modes because of short closing of window.
The status of the ***window open mode*** is available on CO Nr. 19/, *Status* ***window open mode***.

### Valve behaviour in window open mode

This parameter is used to indicate if the valve must be closed in ***window open mode***.
If choosing "No incidence", opening / closing windows will have no incidence on the valve.
If choosing "Close valve if window open", the output voltage will be set to 0V for analog 0-10V valve or to 0% pulse ratio for PWM valve, once device is in ***window open mode***.

### Relay behaviour in window open mode

This parameter is used to indicate if the relay must be switched OFF in ***window open mode***.
If choosing "No incidence", opening / closing windows will have no incidence on the relay.
If choosing "Switch relay OFF if window open", the relay will be switched OFF once device is in ***window open mode***.

### Fan speed in window open mode

This parameter is used to define the fan speed to switch once device is in ***window open mode***.

For more detailed information about the single status CO's refer to the communication object description.

## 3.2.7 *Forcing mode* parameters

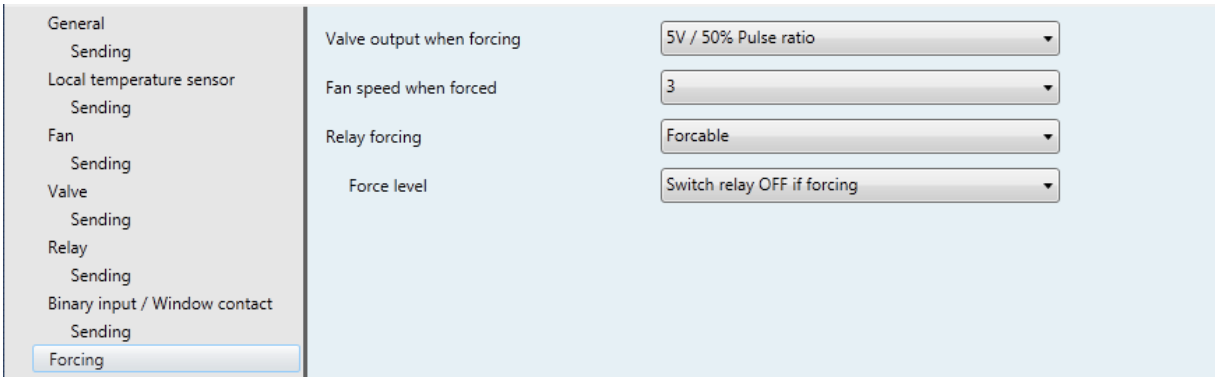



Figure 18: ***Forcing mode*** parameters

CO Nr. 1/, ***Forcing mode***, can be used to put the device in a predefined state.
***Forcing mode*** will act on valve output and fan speed. This may be used for example for maintenance, in order to drive the valves at least once / month in order to avoid valves to stuck, or to close all valves if working on the pipes.
It can also be used to protect a whole building against frost or overheating.
Also, the relay can be forced.
***Forcing mode*** has highest priority against all other working modes. Only ***Handmode*** can override ***forcing mode***.
A re-start of device (power return, programing, …) will de-activate ***forcing mode*** of device.
Once ***forcing mode*** is deactivated, device works in same way as before forcing.

**Valve output when forcing**
This parameter is used to set the valve output value when device is in ***forcing mode***.

**Fan speed when forced**
This parameter is used to set the fan speed when device is in ***forcing mode***.

**Relay forcing**
This parameter is used to indicate if ***forcing mode*** should also act on relay.
If choosing "No incidence", ***forcing mode*** will not act on the relay.
If choosing "Forcable" a new parameter will be displayed in order to define the behaviour of relay if device is in ***forcing mode***.

### Force level
This parameter is used to define the behaviour of the relay if forcable.
If choosing "Switch relay OFF if forcing", the relay will be switched OFF when device is in ***forcing mode***.
If choosing "Switch relay ON if forcing", the relay will be switched ON when device is in ***forcing mode***.

| | |
|---|---|
| Overriding | ***Handmode*** |
| Highest priority | ***Forcing mode*** |
| High priority | ***Window open mode*** |
| Low priority | ***Manual mode*** |
| Lowest priority | ***Automatic mode*** |

Figure 19: Priority of different working modes

## 3.2.8 General sending parameters

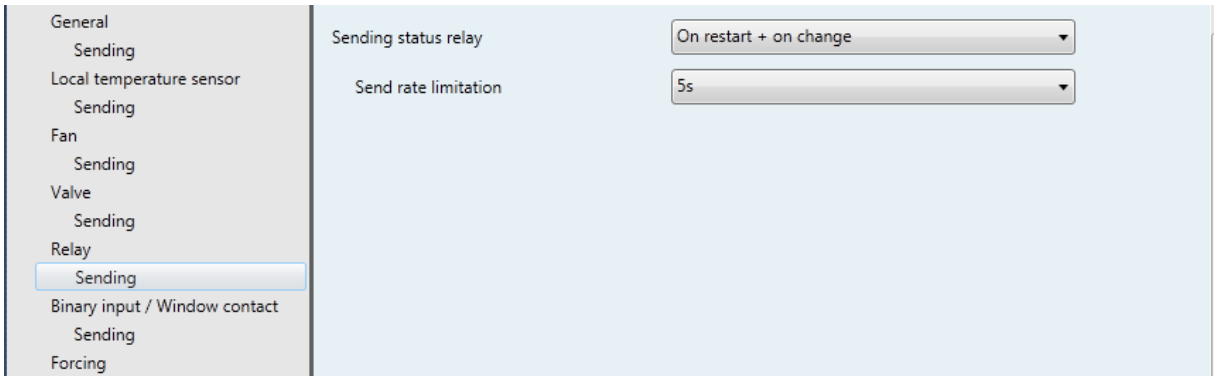



Figure 20: Sending parameters

The parameters of the device are separated in different functional parts:
General, Temperature, Fan, Valve, Relay, Binary input / Window contact, Forcing.

For most of these parts, some status CO's exists. The parameters for setting up the status CO's are accessible in nested parameter sets called *Sending*.
The parameters will only be visible if relevant. So, depending on settings, all parameters for status CO's may not be visible.

Status CO's are sent out by the device. In most cases, the status sending mechanism can be set in following way with parameters:

- "Never": this CO will never be sent on bus.
- "On restart": this CO will be updated on bus after each restart of device (and also after bus failure recovery or after reprogramming device). In order to avoid bus overload after restart or recovery of a complete system, the "delay before starting to send after restart" can be set individually for each device within "general" parameters.
- "On restart + cyclic": All CO's set by parameters to be sent out cyclically will be sent out together in given time delays. This delay can be set by parameters. The cyclical sending will only begin after the "Delay before starting to send after restart" is elapsed.
- "On restart + on change": The CO's set by parameter to be sent out by change will be updated on bus individually on change of associated value. In order to avoid bus overloads due to fast changes, most of this CO's have an individual "send rate limitation" set by parameter. Even on change, the CO with new value will not be sent out until this time is elapsed since last sending of same CO. The sending on change will only begin after the "Delay before starting to send after restart" is elapsed.

For more detailed information about the single status CO's refer to the communication object description.

# 4. Changing the replaceable fuse

**Before changing the internal fuse, unplug connectors for supply, relay and fan from the fancoil actuator**


Remove the terminal cover with a screwdriver


Remove the fuse
Insert a new fuse and replace the terminal cover.
Only replace by fuse of same type!
***Subminiature fuse, 8.5mm, 5A Time-Lag T, 250VAC, IEC 60127-3***

# 5. Technical datas

***Technical Data Art. 1730.02120/55100***

| | |
|---|---|
| **Housing** | 106mm (6 units) x 110 x 60mm |
| Mounting | To be mounted on 35mm DIN Rail |
| Material | UL94-V0 flame retardant polycarbonate |
| Protection degree | IP20 |
| Weight | 0.370kg |
| | |
| **Operating temperature** | -5°C … +45°C |
| | |
| **Terminals** | Pluggable screw terminals, 16A 450V |
| Wire section | 2.5mm2 (14AWG) stranded / 1.5mm2 (16AWG) |
| Cage clamp recess size | 2.8x3.1mm / 1.8x2.6mm |
| Captive screw | M3 /M2 |
| Recommended tightening torque | 0.5Nm (4.5in.lbs.) / 0.25Nm (2.3in.lbs.) |
| | |
| **Supply** | 230VAC, 50/60Hz |
| Replaceable fuse | Subminiature fuse, 8.5mm, 5A Time-Lag T, 250VAC, IEC 60127-3 |
| | The same fuse supplies device and all outputs! |
| | |
| **KNX** | 10mA current consumption from bus |
| | |
| **Window contact input** | NO or NC dry contact |
| Voltage supplied from device to window contact | 24VDC, max. 6mA (4k Ohm Internal) |
| Cabling | Twisted pair, Maximum 30m |
| | |
| **Temperature sensor input** | Temperature sensor from DATEC Electronic<br>Art. 1630.03121/56XXX (XXX length in cm) |
| Cabling | Twisted pair, Maximum 30m |
| Voltage supplied from device to temperature sensor | 24VDC, modulated |
| | |
| **Relay outputs** | 230VAC |
| Total current of all outputs | Maximum 5A |
| Contact rating | 5A, 250VAC |
| Contact material | AgNi |
| Maximum switching power | 1250VA |
| Dielectric strength | 1000VAC 1minute between open contacts |
| | |
| **Valve output** | 24VAC PWM or 0-10VDC modulating |
| Voltage supplied from device to valve | 24VAC, max. 4.5VA nominal, overload / short circuit protected |
| Type of valve | 24VAC PWM or 0-10VDC modulating |
| 24VAC PWM output pulse ratio | 0 – 100% |
| 24VAC PWM Period | 15seconds … 1 Hour |
| 0-10VDC modulating output | 0-10VDC, max. 10mA |
| cabling | Maximum 3m |
| | |
| **EMC** | EN 61000-4-2 |
| | EN 61000-4-3 |
| | EN 61000-4-4 |
| | EN 61000-4-5 |
| | EN 61000-4-6 |
| | EN 61000-4-11 |