KNX
Product documentation

GIRA

Issue:
01.04.2020

# Gira G1 PoE
# Gira G1 24 V
# Gira G1 230 V
# Order no. 2069 00
# Order no. 2077 00
# Order no. 2067 00

## Table of Contents

# 1. Range of functions

### KNX functions of the Gira G1

- 150 channel functions can be implemented in up to 6 function folders (max. 25 channel functions per folder) with the Gira G1.
- As required, the KNX channels can be configured to the functions "Switching", "Trigger ON/OFF", "Press / release ON/OFF", "Dimming brightness value", "Dimming relative", "Dimming RGB", "Dimming RGBW", "Dimming Tunable White", "Blind / shutter step move", "Blind positioning", "Shutter positioning", "Room temperature","Room temperature ON/OFF", "Sauna temperature", "Sauna temperature ON/OFF", "Air conditioning", "Scene extension", "Value transmitter", "Status display", "IP camera", "Open URL", "Audio Control", "Audio control with media data" and "Audio control with playlist", or alternatively to "no function".
- Each function folder and each channel function can be assigned a function icon.
- KNX channels have different status elements, depending on their function:
  - Status text
  - Status value
  - Status ring
  - Status LED
  - Background colour of buttons
  - Operable scale
- Depending on their function, buttons on the user interface can be displayed and evaluated as buttons (single-surface principle) or as rockers (double-surface principle).
- Two temperature values (outdoor temperature and room temperature), as well as time and date, can be displayed in the status bar of the Gira G1.
- The room temperature detection makes it possible for temperature values to be received and/or cyclically queried through the KNX.
- The time and date can be received through the KNX and/or transmitted to the same.
- Received temperature values can be compared.

### Functions of the room temperature controller

- There are four operating modes that can be activated (Comfort, Standby, Night and Frost/Heat Protection).
- Each operating mode can be assigned setpoint temperature values (for heating and/or cooling).
- Setpoint temperature values are configured with absolute values.
- The comfort extension can be implemented for a parameterisable period of time using the "Presence" button in night or frost/heat protection mode.
- Changeover between heating and cooling mode takes place using the Gira G1 Display or a 1-bit object.
- Operating mode changeover takes place using 1-byte objects in accordance with the KNX specification.
- Various status feedback messages (also KNX compliant) can be configured.
- Frost/heat protection changeover can be carried out via window status.
- Room temperature controller information is shown on the Gira G1 Display.
- The room temperature controller is operated using the Gira G1 Display.
- The room temperature controller distinguishes between the operating modes of "Heating", "Cooling" and "Heating and cooling".
- Various control types can be configured according to the heating or cooling level: PI control (constant or switching PWM) or 2-point control (switching).

- Temporary setpoint offset is possible by using the buttons on the device Display or communication objects (e.g. a controller extension). The setpoint offset may be depicted in the device Display in the form of an operable scale.
- The step width of the setpoint offset can be parameterised (0.1 K / 0.2 K / 0.5 K / 1.0 K).
- The control can be deactivated through a separate 1-bit object.
- The corrected variables can be limited.
- The clipping mode (behaviour of controller when variables = 100%) is adjustable.
- Separate or joint corrected variable output in heating or cooling mode. This results in one or two corrected variable objects per stage.
- Normal or inverted corrected variable output can be parameterised.
- Automatic transmission and cycle time for corrected variable output can be parameterised.

**KNX Secure**

The Gira G1 is compatible with KNX Secure from index level I07 or higher when used in combination with firmware version 3.2 or higher.

KNX Secure provides protection against the manipulation of building automation, and can be configured within the ETS project.

Detailed specialist knowledge is required. A device certificate (FDSK Factory Default Setup Key), which is affixed to the device, is required for secure start-up. While performing installation, removing the certificate from the device and storing it in a safe place is recommended.

Note:
If the Gira G1 is being integrated into KNX Secure systems, the device certificate is required for the initial download when starting up the Gira G1 using ETS.

In addition, the device certificate is required when repeating start-up of devices that were reset to the factory settings.

Where is the device certificate?

- In devices with index status I07 or higher, you will find the device certificate on a sticker on the back of the device.
  Note: In systems with very strict security requirements, the device certificate should be noted elsewhere and be rendered illegible on the device.
- In devices with index status I08 or higher, you will also find the device certificate on the secure card included with the device.

Important:

- In the event of loss of the device certificate, no new start-up of the Gira G1 in KNX Secure systems is possible.
- Please keep the device certificate and the secure card in a safe place.
- The device certificate and the secure card for the installed devices must remain with the customer.

# 2. Software

## 2.1. Information on the software

### ETS configuration and start-up

Configuration and start-up require the start-up software certified by the KNX Association, version ETS5.4 or higher. No product database is available for older versions of the ETS.

From index status I07, the Gira G1 may only be started up through the ETS using KNX long frames.

Please check that

- the "Use reduced bus communication" setting is deactivated in the ETS project settings
- the interface used supports a maximum telegram length greater than 15 bytes.

## 2.2. Software specification

ETS search paths:
- Displays / LCD displays / Gira G1 PoE
- Heating, air conditioning, ventilation / Controller / Gira G1 PoE
- Physical sensors / Temperature / Gira G1 PoE
- Buttons / Push button, general / Gira G1 PoE
- Displays / LCD displays / Gira G1 24 V
- Heating, air conditioning, ventilation / Controller / Gira G1 24 V
- Physical sensors / Temperature / Gira G1 24 V
- Buttons / Push button, general / Gira G1 24 V
- Displays / LCD displays / Gira G1 230 V
- Heating, air conditioning, ventilation / Controller / Gira G1 230 V
- Physical sensors / Temperature / Gira G1 230 V
- Buttons / Push button, general / Gira G1 230 V

Configuration: S-mode standard

### Application for Gira G1:

| No. | Brief description | Name | Version | From mask version |
|---|---|---|---|---|
| 1 | Multi-functional room operating device for visualising and operating various building functions. With integrated room temperature controller (RTC). For use in the KNX system and Gira door communication system. | Gira G1 111012 | 2.0 for ETS5.4 | SystemB IP (57B0) |

This application is designed for configuration and start-up of the Gira G1 from firmware version 2.0.442. Devices with older firmware versions cannot be configured with this application.

## 3. Object table

Number of communication objects: 3042
Number of addresses (max.): 32767
Number of assignments (max.): 32767

### Information on initialisation upon device restart

After a device has been restarted, it may be necessary for communication objects to actively query the value of their group address via the KNX.

Logical uses for active querying of a group address include presentation of controller statuses (e.g. controller blocked status), temperature display and passing on of information from a device clock.

After a device has been restarted, a group address is actively queried if the communication flags "Read upon init" and "Transmit" are set. Communication flags can be set or deleted in the ETS in accordance with the KNX specification.

### Example

Temperature display in the Gira G1 user interface.

If the temperature value is not actively queried following a device restart, a temperature value cannot be displayed in the user interface for the time that it takes for the communication object to be written with a temperature value telegram. During this period, "--.-" is shown at the appropriate point in the Gira G1 status bar.

This can be avoided by setting the "Read upon init" and "Transmit" flags.

The "Read" flag of the transmitting communication object must be set in order to be able to read out the object value.

### 3.1. General

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 1 | Gira G1 - output | Ready | 1 bit | 1.011 status | C, R, T |

Description: Object which returns the state of the device. If the object has a value of "1", the device is ready for operation. A "0" means that the device is not yet ready for operation.

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 2 | Gira G1 - output | Status | 1 byte | 5.010 meter pulses (0..255) | C, R, T |

Description: Object for transmitting the current device state.

| | | |
|---|---|---|
| 00 h | Ready | The Gira G1 is ready for operation. |
| 01 h | Start up | The Gira G1 is starting up. |
| 02 h | Shut down | The Gira G1 will restart. |
| 03 h | Configuration | The Gira G1 is being configured. |
| 04 h...FF h | Reserved | |

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 3 | Gira G1 - input | Restart | 1 bit | 1.015 reset | C, W |

Description: Object for receiving the restart request.
This object is only available if the setting is "Display restart: **Active**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 4 | Device clock - output | Date | 3 bytes | 11.001 date | C, R, T |

Description: Object for transmitting the current date.
This object is only available if the setting is "Time mode: **Gira G1 is time server**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 5 | Device clock - output | Time | 3 bytes | 10.001 time of day | C, R, T |

Description: Object for transmitting the current time.
This object is only available if the setting is "Time mode: **Gira G1 is time server**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 6 | Device clock - input | Accept date | 3 bytes | 11.001 date | C, W |

Description: Object for receiving the current date.
This object is only available if the setting is "Time mode: **Gira G1 is time client**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 7 | Device clock - input | Accept time | 3 bytes | 10.001 time of day | C, W |

Description: Object for receiving the current time.
This object is only available if the setting is "Time mode: **Gira G1 is time client**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 8 | Gira G1 - output | Runtime (s) | 4 bytes | 13.100 time difference (s) | C, R |

Description: Object for transmitting the operating time in seconds. Time since last restart.

| | | | | | | |
|---|---|---|---|---|---|---|
| | 40 | Gira G1 - input | red LED | 1 bit | 1.001 switching | C, W |

Description: Object for setting the red LED. "1" means the red LED is on, "0" means it is off.
When used in conjunction with the "green LED" and "blue LED" communication objects, different colours can be set:

| Colour | red LED | green LED | blue LED |
|---|---|---|---|
| Red | 1 | 0 | 0 |
| Green | 0 | 1 | 0 |
| Blue | 0 | 0 | 1 |
| Yellow | 1 | 0 | 1 |
| Cyan | 0 | 1 | 1 |
| Magenta | 1 | 1 | 0 |
| White | 1 | 1 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 41 | Gira G1 - input | green LED | 1 bit | 1.001 switching | C, W |

Description: Object for setting the green LED. "1" means the green LED is on, "0" means it is off. When used in conjunction with the "red LED" and "blue LED" communication objects, different colours can be set (see above).

| | | | | | | |
|---|---|---|---|---|---|---|
| | 42 | Gira G1 - input | blue LED | 1 bit | 1.001 switching | C, W |

Description: Object for setting the blue LED. "1" means the blue LED is on, "0" means it is off. When used in conjunction with the "red LED" and "green LED" communication objects, different colours can be set (see above).

| | | | | | | |
|---|---|---|---|---|---|---|
| | 50 | Gira G1 - input | Programming mode | 1 bit | 1.001 switching | C, T |

Description: Object for receiving the programming mode. "1" means that programming mode is activated, and "0" means that programming mode is deactivated.
This object is only available if the setting is "Display KNX statuses: **Active**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 51 | Gira G1 - output | Programming mode status | 1 bit | 1.001 switching | C, R, T |

Description: Object for transmitting the programming mode. "1" means that programming mode is activated, and "0" means that programming mode is deactivated.
This object is only available if the setting is "Display KNX statuses: **Active**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 62 | Door communication - input | Floor button | 1 bit | 1.017 trigger | C, W |

Description: Object for receiving a floor-call
This object is only available if the setting is "General/Use floor button in KNX: **Active**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 63 | Door communication - input | Mute ringtone | 1 bit | 1.001 switching | C, W |

Description: Object for muting the ringtone.
This object is only available if the "Gira door communication system" or "SIP door communication system" entry has been selected for the "Door communication type" parameter.

| | | | | | | |
|---|---|---|---|---|---|---|
| | 64 | Door communication - input | Mute ringtone, feedback | 1 bit | 1.001 switching | C, W |

Description: Object for receiving feedback on whether the ringtone is muted.
This object is only available if the "Gira door communication system" or "SIP door communication system" entry has been selected for the "Door communication type" parameter.

| | | | | | | |
|---|---|---|---|---|---|---|
| | 68 | Gira G1 - input | Outdoor temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |

Description: Object for receiving the outdoor temperature.
This object is only available if the setting is "General/Display outdoor temperature in status bar of the Gira G1: **Active**"

### 3.2. Room temperature measurement

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 66 | Room temperature measurement - input | Received temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |

Description: Object for receiving a temperature value.

This object is only available if the setting is "Room temperature measurement/General/Sensor selection: **Received temperature value only** or **internal sensor + received temperature value**

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 67 | Room temperature measurement - output | Actual temperature | 2 bytes | 9.001 temperature (°C) | C, R, T |

Description: Object for transmitting the current actual temperature.

This object is only available if the setting is "General/Use room temperature measurement?: **Active**

### 3.3. Room temperature controller (RTC)

The following objects are only available if the setting is "General/Use room temperature controller?: Active".

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 71 | RTC - input | Setpoint value active operating mode | 2 bytes | 9.001 temperature (°C) | C, W |

Description: 2-byte object for externally specifying a setpoint value. The possible value range is limited in dependence on the operating mode by the parameterised frost protection and/or heat protection temperature. The controller rounds the temperature values received through the object to the specified step width of the setpoint offset. The step width can be configured in the ETS (0.1 K / 0.2 K / 0.5 K / 1.0 K). The temperature value must always be specified in the "°C" format.

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 72 | RTC - output | Setpoint temperature | 2 bytes | 9.001 temperature (°C) | C, W |

Description: 2-byte object for output of the current setpoint temperature value of the control circuit. The possible value range is limited in dependence on the operating mode by the parameterised frost protection and/or heat protection temperature. The temperature value must always be output in the "°C" format

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 73 | RTC - input | Operating mode changeover | 1 byte | 20.102 HVAC mode | C, W |

Description: 1-byte object for changeover of the controller operating mode in accordance with the KNX specification.

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 74 | RTC - output | Controller status | 1 byte | | C, R, T |

Description: 1-byte object with which the controller outputs the current operating mode (e.g. to a controller extension). Corresponds to the non-standard DPT DPT_HVACStatus.

This object is only available if the setting is "Room temperature controller/Variables../Status controller: **Controller in general**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 74 | RTC - output | KNX status operating mode | 1 byte | 20.102 HVACMode | C, R, T |

Description: 1-byte object with which the controller outputs the current operating mode. This object is generally used to ensure that controller extensions are able to display the controller operating mode correctly in the KNX compliant status display. Consequently, this object must be connected with controller extensions if KNX compliant status feedback is configured. Only if "Status controller = KNX compliant".
This object is only available if the setting is "Room temperature controller/Variables.../Status controller: **KNX compliant**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 75 | RTC - input | Presence object | 1 bit | 1.018 assignment | C, R, T |

Description: Presence is transmitted to the controller main unit via a button in the Gira G1 user interface. Absence can also be reported to the Gira G1 by the controller main unit. An active presence is displayed in the user interface.
Polarity: presence = "1", no presence = "0".
This object is only available if the setting is "Room temperature controller/Controller functionality/Presence detection: **Presence button** or **Presence detector**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 76 | RTC - output | Status presence object | 1 bit | 1.018 assignment | C, W |

Description: 1-bit feedback object for reading presence. Polarity: presence = "1", no presence = "0"
This object is only available if the setting is "Room temperature controller/Controller functionality/Presence detection: **Presence button** or **Presence detector**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 77 | RTC - output | Heating message | 1 bit | 1.002 Boolean | C, R, T |

Description: 1-bit object for receiving messages from the controller main unit as to whether heat energy is requested. Object value = "1": energy request, object value = "0": no energy request.
This object is only available if the setting is "Room temperature controller/General/Operating mode: **Heating**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 78 | RTC - output | Cooling message | 1 bit | 1.002 Boolean | C, R, T |

Description: 1-bit object for receiving messages from the controller main unit as to whether cooling energy is requested. Object value = "1": energy request, object value = "0": no energy request.
This object is only available if the setting is "Room temperature controller/General/Operating mode: **Cooling**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 82 | RTC - input | Window status | 1 bit | 1.019 window/door | C, W |

Description: 1-bit object for coupling window contacts.
Polarity: window opened = "1", window closed = "0".

| | | | | | | |
|---|---|---|---|---|---|---|
| | 83 | RTC - output | Window status feedback | 1 bit | 1.019 window/door | C, R, T |

Description: 1-bit feedback object for reading the window status. Polarity: window opened = "1", window closed = "0"

| | | | | | |
|---|---|---|---|---|---|
| 84 | RTC - input | Forced object operating mode | 1 byte | 20.102 HVAC mode | C, W |

Description: 1-byte object for forced changeover (top priority) of the controller operating mode in accordance with the KNX specification.

| | | | | | |
|---|---|---|---|---|---|
| 85 | RTC - output | Status message additional | 1 byte | | C, R, T |

Description: 1-byte object with which the controller outputs the current extended operating mode (e. g. to a controller extension). This object is only available if the setting is "Room temperature controller/Variables../Status controller: **Controller in general**".

| | | | | | |
|---|---|---|---|---|---|
| 85 | RTC - output | KNX status forced operating mode | 1 byte | 20.102 HVAC mode | C, R, T |

Description: 1-byte object with which the controller outputs the operating mode in the event of positively-driven operation. This object is generally used to ensure that controller extensions are able to display the controller operating mode correctly in the KNX compliant status display. Consequently, this object must be connected with controller extensions if KNX compliant status feedback is configured.

This object is only available if the setting is "Room temperature controller/Variables../Status controller: **KNX compliant**".

| | | | | | |
|---|---|---|---|---|---|
| 86 | RTC - output | Heating variable | 1 byte | 5.001 percent (0..100%) | C, R, T |

Description: 1-byte object for outputting the constant variable of the heating mode.

This object is only available if the type of control is parameterised as "**Constant PI control**".

| | | | | | |
|---|---|---|---|---|---|
| 86 | RTC - output | Heating/cooling variable | 1 byte | 5.001 percent (0..100%) | C, T |

Description: 1-byte object for outputting the combined constant variable of the heating and cooling mode.

This object is only available if the variables for heating and cooling mode are to be output to a common object (parameter-dependent). In addition, the type of control must be parameterised as "**Constant PI control**".

| | | | | | |
|---|---|---|---|---|---|
| 87 | RTC - output | Heating variable | 1 bit | 1.001 switching | C, R, T |

Description: 1-bit object for outputting the switching variable of the heating mode.

This object is only available if the settings are as follows:

- Room temperature controller/General/Operating mode: **Heating** or **Heating and cooling**
- Room temperature controller/General/Type of heating control: **switching 2-point control**

| | | | | | |
|---|---|---|---|---|---|
| 87 | RTC - output | Heating/cooling variable | 1 bit | 1.001 switching | C, T |

Description: 1-byte object for outputting the combined constant variable of the heating and cooling mode.

This object is only available if the settings are as follows:

- Room temperature controller/General/Operating mode: **Heating and cooling**
- Room temperature controller/General/Type of heating control: **switching 2-point control**

| | | | | | | |
|---|---|---|---|---|---|---|
| | 87 | RTC - output | Heating/cooling variable (PWM) | 1 bit | 1.001 switching | C, T |

Description: 1-bit object for outputting the combined PWM variable of the heating and cooling mode.
This object is only available if the settings are as follows:
- Room temperature controller/General/Operating mode:**Heating and cooling**
- Room temperature controller/General/Type of heating control: **switching PI control (PWM)**

| | | | | | | |
|---|---|---|---|---|---|---|
| | 87 | RTC - output | Heating variable (PWM) | 1 bit | 1.001 switching | C, W |

Description: 1-bit object for outputting the PWM variable of the heating mode.
This object is only available if the settings are as follows:
- Room temperature controller/General/Operating mode: **Heating**
- Room temperature controller/General/Type of heating control: **switching PI control (PWM)**

| | | | | | | |
|---|---|---|---|---|---|---|
| | 88 | RTC - output | Cooling variable | 1 byte | 5.001 percent (0..100%) | C, R, T |

Description: 1-byte object for outputting the constant variable of the cooling mode.
This object is only available if the settings are as follows:
- Room temperature controller/General/Operating mode: **Heating and cooling** or **Cooling**
- Room temperature controller/General/Type of cooling control: **Constant PI control** or **switching PI control (PWM)**

| | | | | | | |
|---|---|---|---|---|---|---|
| | 89 | RTC - output | Cooling variable | 1 bit | 1.001 switching | C, R, T |

Description: 1-bit object for outputting the PWM variable of the cooling mode.
This object is only available if the settings are as follows:
- Room temperature controller/General/Operating mode: Cooling or Heating and cooling
- Room temperature controller/General/Type of cooling control: switching 2-point control

| | | | | | | |
|---|---|---|---|---|---|---|
| | 89 | RTC - output | Cooling variable (PWM) | 1 bit | 1.001 switching | C, T |

Description: 1-bit object for outputting the switching variable of the cooling mode.
This object is only available if the settings are as follows:
- Room temperature controller/General/Operating mode: Cooling or Heating and cooling
- Room temperature controller/General/Type of cooling control: **switching PI control (PWM)**

| | | | | | | |
|---|---|---|---|---|---|---|
| | 90 | RTC - input | Block controller operation | 1 bit | 1.002 Boolean | C, W |

Description: 1-bit object to deactivate controller operation.
Polarity: operation blocked = "1", operation released = "0".
This object is only available if the setting is "Room temperature controller/Controller functionality/Controller operation lockable: **via bus**"

| | | | | | | |
|---|---|---|---|---|---|---|
| | 91 | RTC - output | Status block controller operation | 1 bit | 1.002 Boolean | C, R, T |

Description: 1-bit object to query whether controller operation is blocked.
Polarity: controller deactivated = "1", controller activated = "0".
This object is only available if the setting is "Room temperature controller/Controller functionality/Controller operation lockable: **via bus**"

| | | | | | |
|---|---|---|---|---|---|
| 92 | RTC - input | Variable limiting | 1 bit | 1.002 Boolean | C, W |

Description: 1-bit object for activating or deactivating the variable limiting of the control circuit.
Polarity: limitation activated = "1", limitation deactivated = "0".
Room temperature controller/Variables../Variable limiting: can be activated via object

| | | | | | |
|---|---|---|---|---|---|
| 93 | RTC - output | Status variable limitation | 1 bit | 1.002 Boolean | C, R, T |

Description: 1-bit object for querying whether the variable limitation of the control circuit is active.
Polarity: limitation activated = "1", limitation deactivated = "0".
This object is only available if the setting is "Room temperature controller/Variables../Variable limiting: **can be activated via object**"

| | | | | | |
|---|---|---|---|---|---|
| 94 | RTC - input | Block controller | 1 bit | 1.002 Boolean | C, W |

Description: 1-bit object to deactivate the controller (activation of dew-point mode).
Polarity: controller deactivated = "1", controller activated = "0".
This object is only available if the setting is "Room temperature controller/Controller functionality/Switch off controller: **active**"

| | | | | | |
|---|---|---|---|---|---|
| 95 | RTC - output | Status block controller | 1 bit | 1.002 Boolean | C, R, T |

Description: 1-bit object to query whether the controller is blocked.
Polarity: controller deactivated = "1", controller activated = "0".
This object is only available if the setting is "Room temperature controller/Controller functionality/Switch off controller: **active**"

| | | | | | |
|---|---|---|---|---|---|
| 96 | RTC - output | KNX status | 2 bytes | 22.101 RHCC status | C, R, T |

Description: 2-byte object with which the controller displays elementary basic functions in a KNX harmonised manner.
This object is only available if the setting is "Room temperature controller/Variables../Status controller = **KNX compliant**".

| | | | | | |
|---|---|---|---|---|---|
| 97 | RTC - input | Heating / cooling toggle | 1 bit | 1.100 heating/cooling | C, W |

Description: 1-bit object for switching between heating and cooling.
Polarity: cooling = "0", heating= "1".
This object is only available if the setting is "Room temperature controller/General/Operating mode: **Heating and cooling**"

| | | | | | |
|---|---|---|---|---|---|
| 98 | RTC - output | Status heating/cooling | 1 bit | 1.100 heating/cooling | C, R, T |

Description: 1-bit for querying whether heating or cooling is taking place.
Polarity: cooling = "0", heating= "1".
This object is only available if the setting is "Room temperature controller/General/Operating mode: **Heating and cooling**"

The following objects are described using the example of channel 1. The objects for the other channels are defined accordingly, with the object number shifted and the object name changed.

### 3.4. "Switching" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching ON/OFF. | | | | | |
| 101 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for switching ON/OFF. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.5. "Trigger ON/OFF" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Trigger | 1 bit | 1.017 trigger | C, T |
| Description: 1-bit object for triggering various actions. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.6. "Press / release ON/OFF" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Press and release | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching. Telegrams are sent in accordance with the configuration. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.7. "Dimming brightness value" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching. Telegrams are sent in accordance with the configuration. | | | | | |
| 101 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for switching. | | | | | |
| 102 | Channel n[1] | Brightness value | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the brightness. 100% is the maximum. | | | | | |
| 103 | Channel n[1] | Feedback brightness value | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the brightness value. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.8. “Dimming relative” function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching. Telegrams are sent in accordance with the configuration. | | | | | |
| 101 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for switching. | | | | | |
| 102 | Channel n[1] | Brightness value | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the brightness. 100% is the maximum. | | | | | |
| 103 | Channel n[1] | Feedback brightness value | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the brightness value. | | | | | |
| 104 | Channel n[1] | Dimming | 4 bits | 3.007 dimmer step | C, T |
| Description: 4-bit object for gradual dimming in accordance with the KNX specification | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.9. "Dimming RGB" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching. Telegrams are sent in accordance with the configuration. | | | | | |
| 101 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for switching. | | | | | |
| 102 | Channel n[1] | Brightness value | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the brightness. 100% is the maximum. | | | | | |
| 103 | Channel n[1] | Feedback brightness value | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the brightness value. | | | | | |
| 104 | Channel n[1] | Red percentage | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for specifying the red colour value between 0% and 100%. | | | | | |
| 105 | Channel n[1] | Feedback red percentage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the red percentage. | | | | | |
| 106 | Channel n[1] | Green percentage | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for specifying the green colour value between 0% and 100%. | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 107 | Channel n[1] | Feedback green percentage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the green percentage. | | | | | | |
| 108 | Channel n[1] | Blue percentage | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for specifying the blue colour value between 0% and 100%. | | | | | | |
| 109 | Channel n[1] | Feedback blue percentage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the blue percentage. | | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.10. "Dimming RGBW" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching. Telegrams are sent in accordance with the configuration. | | | | | |
| 101 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for switching. | | | | | |
| 102 | Channel n[1] | Brightness value | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the brightness. 100% is the maximum. | | | | | |
| 103 | Channel n[1] | Feedback brightness value | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the brightness value. | | | | | |
| 104 | Channel n[1] | Red percentage | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for specifying the red colour value between 0% and 100%. | | | | | |
| 105 | Channel n[1] | Feedback red percentage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the red percentage. | | | | | |
| 106 | Channel n[1] | Green percentage | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for specifying the green colour value between 0% and 100%. | | | | | |
| 107 | Channel n[1] | Feedback green percentage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the green percentage. | | | | | |
| 108 | Channel n[1] | Blue percentage | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for specifying the blue colour value between 0% and 100%. | | | | | |
| 109 | Channel n[1] | Feedback blue percentage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the blue percentage. | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 110 | Channel n[1] | White percentage | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the white percentage. | | | | | | |
| | 111 | Channel n[1] | Feedback white percentage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the white percentage. | | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.11. "Dimming Tunable White" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching. Telegrams are sent in accordance with the configuration. | | | | | |
| 101 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for switching. | | | | | |
| 102 | Channel n[1] | Brightness value | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the brightness. 100% is the maximum. | | | | | |
| 103 | Channel n[1] | Feedback brightness value | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for the brightness value. | | | | | |
| 104 | Channel n[1] | Colour temperature | 2 bytes | 9.002 temperature difference (K) | C, T |
| Description: 2-byte object for setting the colour temperature value. | | | | | |
| 105 | Channel n[1] | Feedback colour temperature | 2 bytes | 9.002 temperature difference (K) | C,W,T,U |
| Description: 2-byte feedback object for reading the colour temperature value. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.12. "Blind / shutter step move" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Curtain short-term operation (stop) | 1 bit | 1.007 step | C, T |
| Description: 1-bit object for activating short-term operation | | | | | |
| 101 | Channel n[1] | Curtain long-term operation | 1 bit | 1.008 up/down | C, T |
| Description: 1-bit object for activating long-term operation | | | | | |
| 102 | Channel n[1] | Feedback drive movement | 1 bit | 1.002 Boolean | C,W,T,U |
| Description: 1-bit feedback object for reading the current drive movement. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.13. "Blind positioning" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Curtain short-term operation (stop) | 1 bit | 1.007 step | C, T |
| Description: 1-bit object for activating short-term operation | | | | | |
| 101 | Channel n[1] | Curtain long-term operation | 1 bit | 1.008 up/down | C, T |
| Description: 1-bit object for activating long-term operation | | | | | |
| 102 | Channel n[1] | Feedback drive movement | 1 bit | 1.002 Boolean | C,W,T,U |
| Description: 1-bit feedback object for reading the current drive movement. | | | | | |
| 103 | Channel n[1] | Curtain position | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-bit object for setting the curtain position. | | | | | |
| 104 | Channel n[1] | Feedback curtain position | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-bit feedback object for reading the curtain position. | | | | | |
| 105 | Channel n | Slat position | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-bit object for setting the slat position. | | | | | |
| 106 | Channel n[1] | Slats feedback of position | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-bit feedback object for reading the slat position. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.14. "Shutter positioning" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Curtain short-term operation (stop) | 1 bit | 1.007 step | C, T |
| Description: 1-bit object for activating short-term operation | | | | | |
| 101 | Channel n[1] | Curtain long-term operation | 1 bit | 1.008 up/down | C, T |
| Description: 1-bit object for activating long-term operation | | | | | |
| 102 | Channel n[1] | Feedback drive movement | 1 bit | 1.002 Boolean | C,W,T,U |
| Description: 1-bit feedback object for reading the current drive movement. | | | | | |
| 103 | Channel n[1] | Curtain position | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-bit object for setting the curtain position. | | | | | |
| 104 | Channel n[1] | Feedback curtain position | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-bit feedback object for reading the curtain position. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.15. "Room temperature controller" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Actual temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| Description: 2-byte object for reading the actual temperature. | | | | | |
| 101 | Channel n[1] | Setpoint temperature | 2 bytes | 9.001 temperature (°C) | C, T |
| Description: 2-byte object for setting the setpoint temperature. | | | | | |
| 102 | Channel n[1] | Feedback setpoint temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| Description: 2-byte object for reading the current setpoint temperature. | | | | | |
| 104 | Channel n[1] | Status of operating mode | 1 byte | 20.102 HVAC mode | C,W,T,U |
| Description: 1-byte for reading the operating mode in accordance with the KNX specification. | | | | | |
| 104 | Channel n[1] | Status controller | 1 byte | | C,W,T,U |
| Description: 1-byte for reading the operating status. This is based on the Non-DPT DPT_HVACStatus. | | | | | |
| 105 | Channel n[1] | Presence object | 1 bit | 1.002 Boolean | C, T |
| Description: 1-bit object for activating presence mode.<br>Polarity: presence active = "1", presence not active = "0". | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 107 | Channel n[1] | Heating message | 1 bit | 1.002 Boolean | C,W,T,U |

Description: 1-bit object for reading information as to whether heating is taking place.
Polarity: heating taking place = "1", heating not taking place = "0"

| | | | | | |
|---|---|---|---|---|---|
| 108 | Channel n[1] | Cooling message | 1 bit | 1.002 Boolean | C,W,T,U |

Description: 1-bit object for reading information as to whether cooling is taking place.
Polarity: cooling taking place = "1", cooling not taking place = "0"

| | | | | | |
|---|---|---|---|---|---|
| 109 | Channel n[1] | Heating / cooling toggle | 1 bit | 3.007 heating/cooling | C, T |

Description: 1-bit object for switching between the heating and cooling operating modes.
Polarity: heating = "1", cooling = "0".

| | | | | | |
|---|---|---|---|---|---|
| 110 | Channel n[1] | Feedback heating/cooling | 1 bit | 3.007 heating/cooling | C,W,T,U |

Description: 1-bit feedback object for the heating and cooling operating modes.
Polarity: heating = "1", cooling = "0".

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.16. "Room temperature controller ON/OFF" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Actual temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |

Description: 2-byte object for reading the actual temperature.

| | | | | | |
|---|---|---|---|---|---|
| 101 | Channel n[1] | Setpoint temperature | 2 bytes | 9.001 temperature (°C) | C, T |

Description: 2-byte object for setting the setpoint temperature.

| | | | | | |
|---|---|---|---|---|---|
| 102 | Channel n[1] | Feedback setpoint temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |

Description: 2-byte object for reading the current setpoint temperature.

| | | | | | |
|---|---|---|---|---|---|
| 104 | Channel n[1] | Status of operating mode | 1 byte | 20.102 HVAC mode | C,W,T,U |

Description: 1-byte for reading the operating mode in accordance with the KNX specification.

| | | | | | |
|---|---|---|---|---|---|
| 104 | Channel n[1] | Status controller | 1 byte | | C,W,T,U |

Description: 1-byte for reading the operating status. This is based on the Non-DPT DPT_HVACStatus.

| | | | | | |
|---|---|---|---|---|---|
| 105 | Channel n[1] | Presence object | 1 bit | 1.002 Boolean | C, T |

Description: 1-bit object for activating presence mode.
Polarity: presence active = "1", presence not active = "0".

| | | | | | |
|---|---|---|---|---|---|
| 107 | Channel n[1] | Heating message | 1 bit | 1.002 Boolean | C,W,T,U |

Description: 1-bit object for reading information as to whether heating is taking place.
Polarity: heating taking place = "1", heating not taking place = "0"

| | | | | | |
|---|---|---|---|---|---|
| 108 | Channel n[1] | Cooling message | 1 bit | 1.002 Boolean | C,W,T,U |

Description: 1-bit object for reading information as to whether cooling is taking place.
Polarity: cooling taking place = "1", cooling not taking place = "0"

| | | | | | |
|---|---|---|---|---|---|
| 109 | Channel n[1] | Heating / cooling toggle | 1 bit | 3.007 heating/cooling | C, T |

Description: 1-bit object for switching between the heating and cooling operating modes.
Polarity: heating = "1", cooling = "0".

| | | | | | |
|---|---|---|---|---|---|
| 110 | Channel n[1] | Feedback heating/cooling | 1 bit | 3.007 heating/cooling | C,W,T,U |

Description: 1-bit feedback object for the heating and cooling operating modes.
Polarity: heating = "1", cooling = "0".

| | | | | | |
|---|---|---|---|---|---|
| 111 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |

Description: 1-bit object for switching the heating.
Polarity: heating on = "1", heating off = "0"

| | | | | | |
|---|---|---|---|---|---|
| 112 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |

Description: 1-bit feedback object for reading whether the heating is switched on.
Polarity: heating on = "1", heating off = "0"

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.17. "Sauna temperature controller" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Actual temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |

Description: 2-byte object for reading the actual temperature.

| | | | | | |
|---|---|---|---|---|---|
| 101 | Channel n[1] | Setpoint temperature | 2 bytes | 9.001 temperature (°C) | C, T |

Description: 2-byte object for setting the setpoint temperature.

| | | | | | |
|---|---|---|---|---|---|
| 102 | Channel n[1] | Feedback setpoint temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |

Description: 2-byte object for reading the current setpoint temperature.

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.18. "Sauna temperature controller ON/OFF" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Actual temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| Description: 2-byte object for reading the actual temperature. | | | | | |
| 101 | Channel n[1] | Setpoint temperature | 2 bytes | 9.001 temperature (°C) | C, T |
| Description: 2-byte object for setting the setpoint temperature. | | | | | |
| 102 | Channel n[1] | Feedback setpoint temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| Description: 2-byte object for reading the current setpoint temperature. | | | | | |
| 103 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching the sauna.<br>Polarity: sauna on = "1", sauna off = "0" | | | | | |
| 104 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for reading whether the sauna is switched on.<br>Polarity: sauna on = "1", sauna off = "0" | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.19. "Air conditioning" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Actual temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| Description: 2-byte object for reading the actual temperature. | | | | | |
| 101 | Channel n[1] | Setpoint temperature | 2 bytes | 9.001 temperature (°C) | C, T |
| Description: 2-byte object for setting the setpoint temperature. | | | | | |
| 102 | Channel n[1] | Feedback setpoint temperature | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| Description: 2-byte object for reading the current setpoint temperature. | | | | | |
| 103 | Channel n[1] | Switching | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for switching the air conditioning.<br>Polarity: air conditioning on = "1", air conditioning off = "0" | | | | | |
| 104 | Channel n[1] | Feedback switching | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for reading whether the air conditioning is switched on.<br>Polarity: air conditioning on = "1", air conditioning off = "0" | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 105 | Channel n[1] | Operating mode changeover | 1 byte | 20.105 HVAC control mode | C, T |

Description: 1-byte object for switching the operating mode in accordance with the KNX specification.

| | | | | | |
|---|---|---|---|---|---|
| 106 | Channel n[1] | Status of operating mode | 1 byte | 20.105 HVAC control mode | C,W,T,U |

Description: 1-byte for reading the operating mode in accordance with the KNX specification.

| | | | | | |
|---|---|---|---|---|---|
| 107 | Channel n[1] | Fan stage | 1 byte | 5.010 meter pulses (0..255) | C, T |

Description: 1-byte object for setting the fan stage.
This object is only available if the settings are as follows:
- Show fan stages: **active**
- Use percentages**: inactive**

| | | | | | |
|---|---|---|---|---|---|
| 107 | Channel n[1] | Fan speed | 1 byte | 5.001 percent (0..100%) | C, T |

Description: 1-byte object for setting the fan speed.
This object is only available if the settings are as follows:
- Show fan stages: **active**
- Use percentages**: active**

| | | | | | |
|---|---|---|---|---|---|
| 108 | Channel n[1] | Feedback fan stage | 1 byte | 5.010 meter pulses (0..255) | C,W,T,U |

Description: 1-byte feedback object for reading the fan stage.
This object is only available if the settings are as follows:
- Show fan stages: **active**
- Use percentages: **inactive**

| | | | | | |
|---|---|---|---|---|---|
| 109 | Channel n[1] | Feedback fan speed | 1 byte | 5.001 percent (0..100%) | C,W,T,U |

Description: 1-byte feedback object for reading the fan speed.
This object is only available if the settings are as follows:
- Show fan stages: **active**
- Use percentages: **active**

| | | | | | |
|---|---|---|---|---|---|
| 109 | Channel n[1] | Horizontal stage | 1 byte | 5.010 meter pulses (0..255) | C, T |

Description: 1-byte object for setting the horizontal stage.
This object is only available if the settings are as follows:
- Show stages horizontal adjustment: **active**
- Use percentages**: inactive**

| | | | | | |
|---|---|---|---|---|---|
| 109 | Channel n[1] | Horizontal position | 1 byte | 5.001 percent (0..100%) | C, T |

Description: 1-byte object for setting the horizontal position.
This object is only available if the settings are as follows:
- Show stages horizontal adjustment: **active**
- Use percentages**: active**

| | | | | | |
|---|---|---|---|---|---|
| 110 | Channel n[1] | Feedback horizontal stage | 1 byte | 5.010 meter pulses (0..255) | C,W,T,U |

Description: 1-byte feedback object for reading the horizontal stage.
This object is only available if the settings are as follows:
- Show stages horizontal adjustment: **active**
- Use percentages: **inactive**

| | | | | | |
|---|---|---|---|---|---|
| 110 | Channel n[1] | Feedback horizontal position | 1 byte | 5.001 percent (0..100%) | C,W,T,U |

Description: 1-byte feedback object for reading the horizontal position.
This object is only available if the settings are as follows:
- Show stages horizontal adjustment: **active**
- Use percentages: **active**

| | | | | | |
|---|---|---|---|---|---|
| 111 | Channel n[1] | Horizontal stop/move | 1 bit | 1.001 switching | C, T |

Description: 1-bit object for stopping or moving.
Polarity: stop = "0", move = "1"
This object is only available if the settings are as follows:
- Show horizontal adjustment stop/move: **active**
- Display stages?: **inactive**

| | | | | | |
|---|---|---|---|---|---|
| 112 | Channel n[1] | Feedback horizontal stop/ move | 1 bit | 1.001 switching | C,W,T,U |

1-bit feedback object for reading.
Polarity: stop = "0", move = "1"
This object is only available if the settings are as follows:
- Show horizontal adjustment stop/move: **active**
- Display stages?: **inactive**

| | | | | | |
|---|---|---|---|---|---|
| 113 | Channel n[1] | Vertical stage | 1 byte | 5.010 meter pulses (0..255) | C, T |

Description: 1-byte object for setting the vertical stage.
This object is only available if the settings are as follows:
- Display vertical stages: **active**
- Use percentages**: inactive**

| | | | | | |
|---|---|---|---|---|---|
| 113 | Channel n[1] | Vertical stage | 1 byte | 5.001 percent (0..100%) | C, T |

Description: 1-byte object for setting the vertical position.
This object is only available if the settings are as follows:
- Display vertical stages: **active**
- Use percentages**: active**

| | | | | | |
|---|---|---|---|---|---|
| 114 | Channel n[1] | Feedback vertical stage | 1 byte | 5.010 meter pulses (0..255) | C,W,T,U |

Description: 1-byte feedback object for reading the vertical stage.
This object is only available if the settings are as follows:
- Display vertical stages: **active**
- Use percentages: **inactive**

| | | | | | |
|---|---|---|---|---|---|
| 114 | Channel n[1] | Feedback vertical stage | 1 byte | 5.001 percent (0..100%) | C,W,T,U |

Description: 1-byte feedback object for reading the vertical position.
This object is only available if the settings are as follows:
- Display vertical stages: **active**
- Use percentages: **active**

| | | | | | |
|---|---|---|---|---|---|
| 115 | Channel n[1] | Vertical stop/move | 1 bit | 1.001 switching | C, T |

Description: 1-bit object for stopping or moving.
Polarity: stop = "0", move = "1"
This object is only available if the settings are as follows:
- Show vertical adjustment stop/move: **active**
- Display stages?: **inactive**

| | | | | | |
|---|---|---|---|---|---|
| 116 | Channel n[1] | Feedback vertical stop/move | 1 bit | 1.001 switching | C,W,T,U |

1-bit feedback object for reading.
Polarity: stop = "0", move = "1"
This object is only available if the settings are as follows:
- Show vertical adjustment stop/move: **active**
- Display stages?: **inactive**

| | | | | | |
|---|---|---|---|---|---|
| 117 | Channel n[1] | Error | 1 bit | 1.005 alarm | C,W,T,U |

Description: 1-bit object for setting when an error has occurred.
Polarity: error = "1", no error = "0"
This object is only available if the setting is "Display error: **active**"

| | | | | | |
|---|---|---|---|---|---|
| 118 | Channel n[1] | Error text | 14 bytes | 16.000 character (ASCII) | C,W,T,U |

14-byte object for setting an error text.
This object is only available if the setting is "Display error text: **active**"

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.20. "Scene extension" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Scene extension | 1 byte | 18.001 scene control | C, T |

Description: 1-byte object for triggering or learning a scene in accordance with the KNX specification.

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.21. "Value transmitter" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Value transmitter (0...100%) | 1 byte | 5.001 percent (0..100%) | C, T |
| 100 | Channel n[1] | Value transmitter (0...255%) | 1 byte | 5.004 percent (0..255%) | C, T |
| 100 | Channel n[1] | Value transmitter (0...255) | 1 byte | 5.010 meter pulses (0..255) | C, T |
| 100 | Channel n[1] | Value transmitter (-128...127) | 1 byte | 6.010 meter pulses (-128..127) | C, T |
| 100 | Channel n[1] | Temperature value transmitter | 2 bytes | 9.001 temperature (°C) | C, T |
| 100 | Channel n[1] | Value transmitter (0...255) | 1 byte | 5.010 meter pulses (0..255) | C, T |
| 100 | Channel n[1] | Value transmitter (0...65535) | 2 bytes | 7.001 pulses | C, T |
| 100 | Channel n[1] | Value transmitter (0...4294967295) | 4 bytes | 12.001 meter pulses (signed) | C, T |
| 100 | Channel n[1] | Value transmitter (-128...127) | 1 byte | 6.010 meter pulses (-128..127) | C, T |
| 100 | Channel n[1] | Value transmitter (-32768...32767) | 2 bytes | 8.001 pulse difference | C, T |
| 100 | Channel n[1] | Value transmitter (-2147483648... 2147483647) | 4 bytes | 13.001 meter pulses (signed) | C, T |
| 100 | Channel n[1] | Decimal value transmitter (2 byte) | 2 bytes | 9.xxx | C, T |
| 100 | Channel n[1] | Decimal value transmitter (4 byte) | 4 bytes | 14.xxx | C, T |
| 101 | Channel n[1] | Feedback value transmitter (0...100%) | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| 101 | Channel n[1] | Feedback value transmitter (0...255%) | 1 byte | 5.004 percent (0..255%) | C,W,T,U |
| 101 | Channel n[1] | Feedback value transmitter (0...255) | 1 byte | 5.010 meter pulses (0..255) | C,W,T,U |
| 101 | Channel n[1] | Feedback value transmitter (-128...127) | 1 byte | 6.010 meter pulses (-128..127) | C,W,T,U |
| 101 | Channel n[1] | Feedback temperature value transmitter | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| 101 | Channel n[1] | Value transmitter (0...255) | 1 byte | 5.010 meter pulses (0..255) | C,W,T,U |
| 101 | Channel n[1] | Feedback value transmitter (0...65535) | 2 bytes | 7.001 pulses | C,W,T,U |

| | | | | | |
|---|---|---|---|---|---|
| 101 | Channel n[1] | Feedback value transmitter (0...4294967295) | 4 bytes | 12.001 meter pulses (signed) | C,W,T,U |
| 101 | Channel n[1] | Feedback value transmitter (-128...127) | 1 byte | 6.010 meter pulses (-128..127) | C,W,T,U |
| 101 | Channel n[1] | Feedback value transmitter (-32768...32767) | 2 bytes | 8.001 pulse difference | C,W,T,U |
| 101 | Channel n[1] | Feedback value transmitter (-2147483648...2147483647) | 4 bytes | 13.001 meter pulses (signed) | C,W,T,U |
| 101 | Channel n[1] | Feedback decimal value transmitter (2 byte) | 2 bytes | 9.xxx | C,W,T,U |
| 101 | Channel n[1] | Feedback decimal value transmitter (4 byte) | 4 bytes | 14.xxx | C,W,T,U |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.22. "Status display" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Status display | 1 bit | 1.xxx | C,W,T,U |
| 100 | Channel n[1] | Value display (0...100%) | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| 100 | Channel n[1] | Value display (0...360°) | 1 byte | 5.003 angle (degrees) | C,W,T,U |
| 100 | Channel n[1] | Value display (0...255% | 1 byte | 5.004 percent (0..255%) | C,W,T,U |
| 100 | Channel n[1] | Value display (0...255) | 1 byte | 5.010 meter pulses (0..255) | C,W,T,U |
| 100 | Channel n[1] | Value display (-128...127) | 1 byte | 6.010 meter pulses (-128..127) | C,W,T,U |
| 100 | Channel n[1] | Value display (0...65535) | 2 bytes | 7.001 pulses | C,W,T,U |
| 100 | Channel n[1] | Value display (-32768..32767) | 2 bytes | 8.001 pulse difference | C,W,T,U |
| 100 | Channel n[1] | Temperature value display | 2 bytes | 9.001 temperature (°C) | C,W,T,U |
| 100 | Channel n[1] | Brightness value display | 2 bytes | 9.004 lux (lux) | C,W,T,U |
| 100 | Channel n[1] | Value display (0...4294967295) | 4 bytes | 12.001 meter pulses (signed) | C,W,T,U |
| 100 | Channel n[1] | Value display (-2147483648...2147483647) | 4 bytes | 13.001 meter pulses (signed) | C,W,T,U |
| 100 | Channel n[1] | Decimal value display | 4 bytes | 14.xxx | C,W,T,U |
| 100 | Channel n[1] | Text display | 14 bytes | 16.000 character (ASCII) | C,W,T,U |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.23. "IP camera" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Camera | 1 bit | 1.002 Boolean | C, T |

Description: 1-bit object for activating the camera.

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.24. "Open URL" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Open URL active | 1 bit | 1.002 Boolean | C, T |
| Description: 1-bit object for activating the URL call. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.25. “Audio Control” function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Playback | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for starting playback.<br>Polarity: playback active = "1", playback not active = "0". | | | | | |
| 101 | Channel n[1] | Feedback playback | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for transmitting the current playback status.<br>Polarity: playback active = "1", playback not active = "0". | | | | | |
| 102 | Channel n[1] | Volume | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the volume. Volume between 0% (quiet) and 100% (loud). | | | | | |
| 103 | Channel n[1] | Feedback volume | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for reading the current volume. Volume between 0% (quiet) and 100% (loud). | | | | | |
| 104 | Channel n[1] | Mute | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for muting.<br>Polarity: muted = "1", not muted = "0". | | | | | |
| 105 | Channel n[1] | Feedback muting | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for reading the current muting status. | | | | | |
| 106 | Channel n[1] | Previous track | 1 bit | 1.017 trigger | C, T |
| Description: 1-bit object for playing the previous track. | | | | | |
| 107 | Channel n[1] | Next track | 1 bit | 1.017 trigger | C, T |
| Description: 1-bit object for playing the next track. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.26. “Audio control with media data” function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Playback | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for starting playback.<br>Polarity: playback active = "1", playback not active = "0". | | | | | |
| 101 | Channel n[1] | Feedback playback | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for transmitting the current playback status.<br>Polarity: playback active = "1", playback not active = "0". | | | | | |
| 102 | Channel n[1] | Volume | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the volume. Volume between 0% (quiet) and 100% (loud). | | | | | |
| 103 | Channel n[1] | Feedback volume | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for reading the current volume. Volume between 0% (quiet) and 100% (loud). | | | | | |
| 104 | Channel n[1] | Mute | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for muting.<br>Polarity: muted = "1", not muted = "0". | | | | | |
| 105 | Channel n[1] | Feedback muting | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for reading the current muting status. | | | | | |
| 106 | Channel n[1] | Previous track | 1 bit | 1.017 trigger | C, T |
| Description: 1-bit object for playing the previous track. | | | | | |
| 107 | Channel n[1] | Next track | 1 bit | 1.017 trigger | C, T |
| Description: 1-bit object for playing the next track. | | | | | |
| 108 | Channel n[1] | Current track | 14 bytes | 16.001 character (ISO 8859-1) | C,W,T,U |
| Description: 14-byte for reading the title of the track currently being played. | | | | | |
| 109 | Channel n[1] | Current album | 14 bytes | 16.001 character (ISO 8859-1) | C,W,T,U |
| Description: 14-byte for reading the album name of the track currently being played. | | | | | |
| 110 | Channel n[1] | Current artist | 14 bytes | 16.001 character (ISO 8859-1) | C,W,T,U |
| Description: 14-byte for reading the name of the artist of the track currently being played. | | | | | |

[1] The object name can be adapted using the "Description of left structure node" parameter.

### 3.27. "Audio control with playlist" function

| Object | Name | Function | Type | DPT | Flags |
|---|---|---|---|---|---|
| 100 | Channel n[1] | Playback | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for starting playback. Polarity: playback active = "1", playback not active = "0". | | | | | |
| 101 | Channel n[1] | Feedback playback | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for transmitting the current playback status. Polarity: playback active = "1", playback not active = "0". | | | | | |
| 102 | Channel n[1] | Volume | 1 byte | 5.001 percent (0..100%) | C, T |
| Description: 1-byte object for setting the volume. Volume between 0% (quiet) and 100% (loud). | | | | | |
| 103 | Channel n[1] | Feedback volume | 1 byte | 5.001 percent (0..100%) | C,W,T,U |
| Description: 1-byte feedback object for reading the current volume. Volume between 0% (quiet) and 100% (loud). | | | | | |
| 104 | Channel n[1] | Mute | 1 bit | 1.001 switching | C, T |
| Description: 1-bit object for muting. Polarity: muted = "1", not muted = "0". | | | | | |
| 105 | Channel n[1] | Feedback muting | 1 bit | 1.001 switching | C,W,T,U |
| Description: 1-bit feedback object for reading the current muting status. | | | | | |
| 106 | Channel n[1] | Previous track | 1 bit | 1.017 trigger | C, T |
| Description: 1-bit object for playing the previous track. | | | | | |
| 107 | Channel n[1] | Next track | 1 bit | 1.017 trigger | C, T |
| Description: 1-bit object for playing the next track. | | | | | |
| 108 | Channel n[1] | Current track | 14 bytes | 16.001 character (ISO 8859-1) | C,W,T,U |
| Description: 14-byte for reading the title of the track currently being played. | | | | | |
| 109 | Channel n[1] | Current album | 14 bytes | 16.001 character (ISO 8859-1) | C,W,T,U |
| Description: 14-byte for reading the album name of the track currently being played. | | | | | |
| 110 | Channel n[1] | Current artist | 14 bytes | 16.001 character (ISO 8859-1) | C,W,T,U |
| Description: 14-byte for reading the name of the artist of the track currently being played. | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 113 | Channel n[1] | Previous playlist | 1 bit | 1.017 trigger | C, T |

Description: 1-bit object for changing to the previous playlist.

| | | | | | |
|---|---|---|---|---|---|
| 114 | Channel n[1] | Next playlist | 1 bit | 1.017 trigger | C, T |

Description: 1-bit object for changing to the next playlist.

| | | | | | |
|---|---|---|---|---|---|
| 115 | Channel n[1] | Name of current playlist | 14 bytes | 16.001 character (ISO 8859-1) | C,W,T,U |

Description: 14-byte for reading the name of the current playlist.

| | | | | | |
|---|---|---|---|---|---|
| 116 | Channel n[1] | Mix playlist | 1 bit | 1.001 switching | C, T |

Description: 1-bit object for activating mix mode.
Polarity: mix = "1", do not mix = "0".

| | | | | | |
|---|---|---|---|---|---|
| 117 | Channel n[1] | Feedback mix playlist | 1 bit | 1.001 switching | C,W,T,U |

Description: 1-bit feedback object for reading the mix mode.
Polarity: mix = "1", do not mix = "0".

| | | | | | |
|---|---|---|---|---|---|
| 118 | Channel n[1] | Repeat playlist | 1 bit | 1.001 switching | C, T |

Description: 1-bit object for activating repeat mode.
Polarity: repeat = "1", do not repeat = "0".

| | | | | | |
|---|---|---|---|---|---|
| 119 | Channel n[1] | Feedback repeat playlist | 1 bit | 1.001 switching | C,W,T,U |

Description: 1-bit feedback object for reading the repeat mode.
Polarity: repeat = "1", do not repeat = "0".

[1] The object name can be adapted using the "Description of left structure node" parameter.

## 4. Functional description

### Settings in the ETS

KNX functions of the device can be configured in the product database of the Gira G1. These are displayed and operated in the user interface after an application has been downloaded. Alongside the functions of room temperature measurement, touch sensor, room temperature controller and controller extension, which are described in detail in the following chapters, further display and operating functions can be enabled in the ETS parameters. The following functions are configured on the "General" parameter page:

### Display outdoor temperature

The Gira G1 can display an outdoor temperature in the °C format. For this to happen, the temperature value must be received via a KNX telegram. The corresponding "Outdoor temperature" communication object (see Page 10) is enabled in the object list through the "Display outdoor temperature in status bar?" parameter. It is a receiving 2-byte object that has to be connected to the transmitting object, e.g. a KNX weather station, through a group address. The temperature value is displayed in the status bar of the user interface. The unit text for displaying the outdoor temperature is set to °C.

### Display room temperature

Only if room temperature measurement is active can the room temperature be displayed in the status bar of the Gira G1. The temperature value is received through the "Received temperature" communication object and only displayed in the user interface of the Gira G1 if the checkmark is set on the parameter "Display room temperature in status bar of the Gira G1?" on the "General" parameter page. This parameter is inactive and blocked in the configuration as long as room temperature measurement is deactivated in the ETS. The unit text for displaying the room temperature is set to °C.

### Time and date

The time and date shown in the status bar can be set either through the KNX or via the internet. In this event, the Gira G1 acts as either the time client or the time server. The necessary communication objects are made available in the ETS when the "Time mode" parameter is set to "GIRA G1 is time client" or "GIRA G1 is time server".

- If "GIRA G1 is time server" is selected, the time (DPT 10.001) and date (DPT 11.001) are queried from an "NTP server" every 10 minutes and cyclically transmitted to the bus every minute via the "Time" and "Date" communication objects. These values can also be read.
- If "GIRA G1 is time client" is selected, the date and time of the device can be set via the bus. Once the "Accept date" and "Accept time" communication objects have been connected with the objects of a KNX system clock through group addresses, the time (DPT 10.001) and date (DPT 11.001) are received through the KNX. The system clock should transmit current values at regular intervals in order to increase the accuracy of the values displayed.

Only one time server may be assigned to the GIRA G1.

- If "Do not use" is selected, there will be no time and date communication objects available to the GIRA G1. The date and time in the status bar are updated via the NTP server when an internet connection is set up.

## 5. Room temperature measurement

Room temperature measurement can be carried out either using received temperature values, via a temperature sensor module that can be plugged into the Gira G1 (from index status I09) or by means of a combination of both measured values. As soon as the "Use room temperature measurement?" parameter is set to active on the "Room temperature measurement" parameter page, further parameters and the "Received temperature" communication object are enabled. The "Display room temperature in status bar?" parameter can then also be configured on the "General" parameter page. The value of the room temperature measurement is displayed in the status bar of the Gira G1 when the checkmark is active.

KNX devices with an integrated temperature sensor (e.g. touch sensor 3 Komfort or KNX room temperature controller) generally have a 2-byte communication object for transmitting the measured temperature value to the KNX. This 2-byte object is connected with the "Received temperature" communication object through a group address as part of the ETS configuration of the Gira G1.

The received temperature value can be adapted to the environmental conditions at the installation site of the KNX device with integrated temperature sensor by setting a fixed adjustment of the received value of between -10 K and +10 K in the parameters of the ETS. It is advisable to configure the KNX device with integrated temperature sensor in such a way that it cyclically emits the measured temperature value to the KNX.

This ensures that the measured room temperature displayed is as up to date as possible. It is also possible to actively query the temperature value through the KNX at regular intervals. To this end, you can configure a query interval of 0 to 240 minutes, with 0 signifying deactivation of automatic querying.

The function of automatic querying of the temperature value must be approved by the transmitting 2-byte communication object of the KNX device with integrated temperature sensor. The read flag must be placed in the object properties for this purpose.

The following aspects should be taken into account when choosing an installation site for the KNX device with integrated temperature sensor:

- Integration of the temperature sensor in multiple combinations should be avoided, particularly when flush mounted dimmers are installed.
- Do not mount the temperature sensor near electrical devices (to avoid heat exposure).
- The temperature sensor should not be installed near radiators or cooling systems.
- Prevent direct sunlight from hitting the temperature sensor.
- Installing sensors on the inside of an external wall can have a negative effect on temperature measurement.
- Temperature sensors should be installed at least 30 cm away from doors, windows or ventilation systems, and at least 1.5 m above the ground.

Room temperature measurement by the device is independent of the "Room temperature control" function (e.g. for easy measurement and display of a room temperature without control).

### 5.1. Temperature detection and measured value generation

The "Sensor selection" parameter in the "Room temperature measurement" parameter node specifies the sensors though which the room temperature is determined:

- "Internal sensor only"
  The actual temperature is detected by the temperature sensor module plugged into the Gira G1.

  Temperature detection via the temperature sensor module is only supported in the Gira G1 from index status I09.

- "Received temperature value only"
  The actual temperature is only determined by means of a temperature value received through the KNX. In this case, the sensor may be a KNX room temperature controller connected via the 2-byte object "Received temperature" or a touch sensor with temperature detection.
  The Gira G1 can request the current temperature value on a cyclical basis. For this to happen, the "Query time for received temperature value" parameter must be set to a value > "0". The query interval can be parameterised within the limits of 1 minute to 240 minutes.
- "internal sensor and received temperature value"
  The actual temperature is determined both through the temperature sensor module and by means of a temperature value received through the KNX. The "Measured value generation to be received internally" parameter can be used to define the relationship between the internal temperature value and the received temperature value.

### 5.2. Comparison of measured values

In some cases, individual temperature values may need to be compared and adjusted as part of the room temperature detection. For example, adjustment is necessary if the temperature measured by the sensors is continually below or above the actual temperature near the sensor. To establish the temperature deviation, the actual room temperature should be determined by means of a reference measurement using a calibrated temperature gauge.

The "Comparison of received temperature value" parameter in the "Room temperature measurement" parameter node can be used to parameterise a temperature adjustment of between -10.0 K and +10.0 K.

The received value must be raised if the value measured by the sensor is below the actual room temperature. The received value must be lowered if the value measured by the sensor is above the actual room temperature.

The Gira G1 always displays the adjusted temperature value.

### 5.3. Comparison of measured values in case of internal sensor

The comparison and adjustment of measured values from the internal sensor can only be carried out via the Gira G1 user interface. The Gira G1 comparison menu ("System" -> "Compare sensor") is displayed if the "Sensor selection" parameter is set to the value "Internal sensor only" or "internal sensor + received temperature value" in the ETS under "Room temperature measurement" -> "General".

To compare and adjust the internal sensor, measure the room temperature at an appropriate point with an accurate thermometer and note down the value. Now enter the measured value directly through the sensor comparison menu. The G1 will then gradually adapt the measured value collection. This process may take up to 20 minutes. No specific message is displayed when the calibration is complete. Please do not carry out any further calibrations during the 20-minute waiting time as this can cause problems.

# 6. Touch sensor functions

## 6.1. Configuring function folders

Touch sensor functions of the Gira G1 can be created in up to six function folders. Each function folder can be assigned a maximum of 25 channel functions. The various channel functions can execute each of the KNX functions on offer (switching, dimming, blind / shutter, ...) and be located together in a function folder. Every configured function folder is displayed on the Home screen as standard. Display on the Home screen can be deactivated in the parameters of the ETS.

To do so, delete the checkmark on the "Display on Home?" parameter on the "Function folder n" parameter page.

Channel functions that are not shown on the Home screen and for which the function folders are not shown on the Home screen are not mapped in the Gira G1. They cannot be used!

A description text and an icon can be defined on the "Function folder n" parameter page. Both elements are displayed in the user interface within a function folder. The total of all channel functions allocated to this function folder is also displayed. Depending on the view in which the function is operated, a function folder is displayed either in the tile view or in the detail view.

It is possible to assign two descriptions for a function folder in the parameters of the ETS. The "Left structure node" description is accepted and displayed in all places in which function folder allocations can be carried out in the ETS. The "User interface" description should be considered separately and is displayed within the function folder in the Gira G1 user interface after application download.

The layout in which function folders and channel functions are displayed in the user interface depends on the order of the parameters in the ETS. The order can be adapted to suit the Gira G1 user interface at a later date in the "Sort functions" view.

The number of visible tiles depends on the number of configured function folders, channel functions and applications.

## 6.2. Channel functions

A functional KNX channel has display elements and operating buttons in the Gira G1 user interface. Depending on the view in which the function is operated, the display and operating area is displayed either with other channel functions in the tile view or in the detail view.

As soon as the "Function" parameter of a KNX channel is set to "no function", this channel will not have any display elements or buttons. Accordingly, no tile or detail view is reserved for this KNX channel.

Various functional elements of a channel function are brought together in the Gira G1 user interface. The description of the channel function [C] and the function icon [B] are shown alongside the description of the function folder [A]. Status elements are also displayed in the user interface of the Gira G1, depending on the set channel function.

The description of the channel function can be defined in the ETS based on the application. The function icon to be displayed can be chosen from a list of icons when the Gira G1 is configured. The function icon is displayed statically.

The appearance and behaviour of the KNX functions listed below can vary depending on the ETS parameterisation. Colours, icons and labels can be parameterised individually for each function in the ETS.

Some elements are only shown when the corresponding communication object is connected with a group address.

### 6.2.1. Switching

Using the "Switching" channel function makes it possible to emit 1-bit switching telegrams (ON, OFF) via the KNX. When combined with switching actuators, this makes it possible to control lighting systems. In addition, the switching telegram can be used elsewhere to execute any control task complying with the 1-bit data format in the KNX system.

The "Switching" channel function can be executed as a button or rocker function, depending on the ETS parameterisation. The graphic (Fig. 1) shows an example of the button function. The graphic (Fig. 2) shows the rocker configuration.

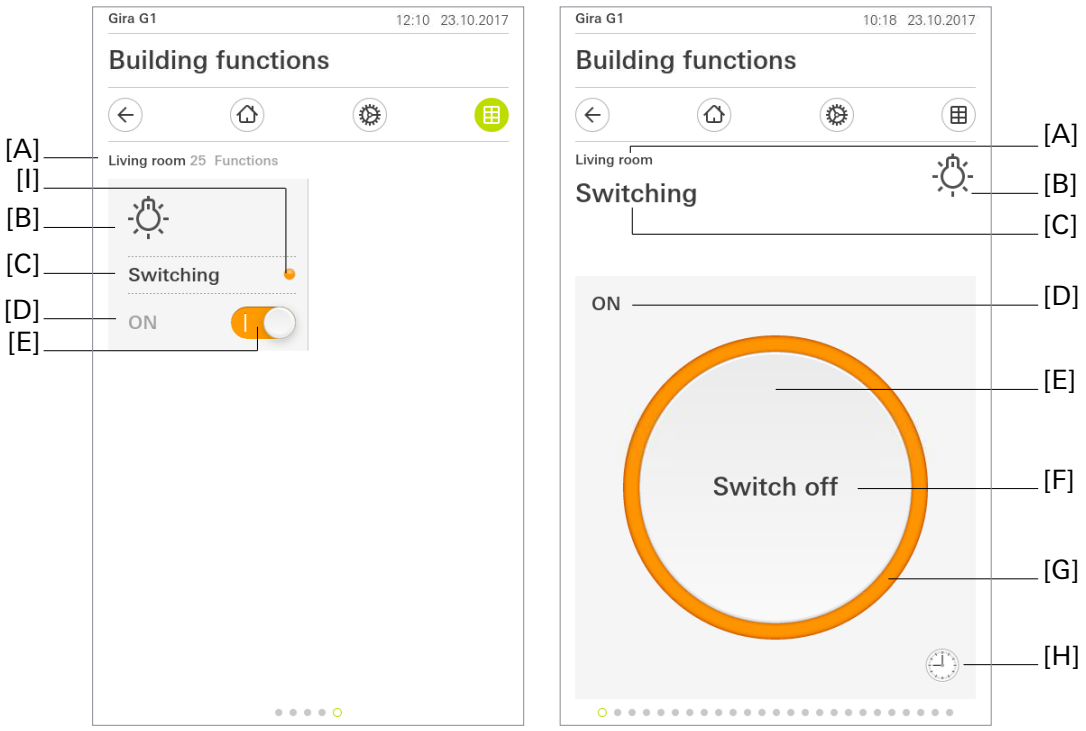


Fig 1: Example of operating and display element of the "Switching" function (button function).

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of "Status text"
[E] Button
[F] Status text
[G] Status ring
[H] "Timer" button
[I] Status LED

In the case of the button function, switching telegrams are sent via a button. The command upon actuating the button (OFF, ON, TOGGLE) is defined in the parameters.
The switching function can be executed in a time-controlled manner using the "Timer" button. You have the option of switching the parameterised function on or off.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

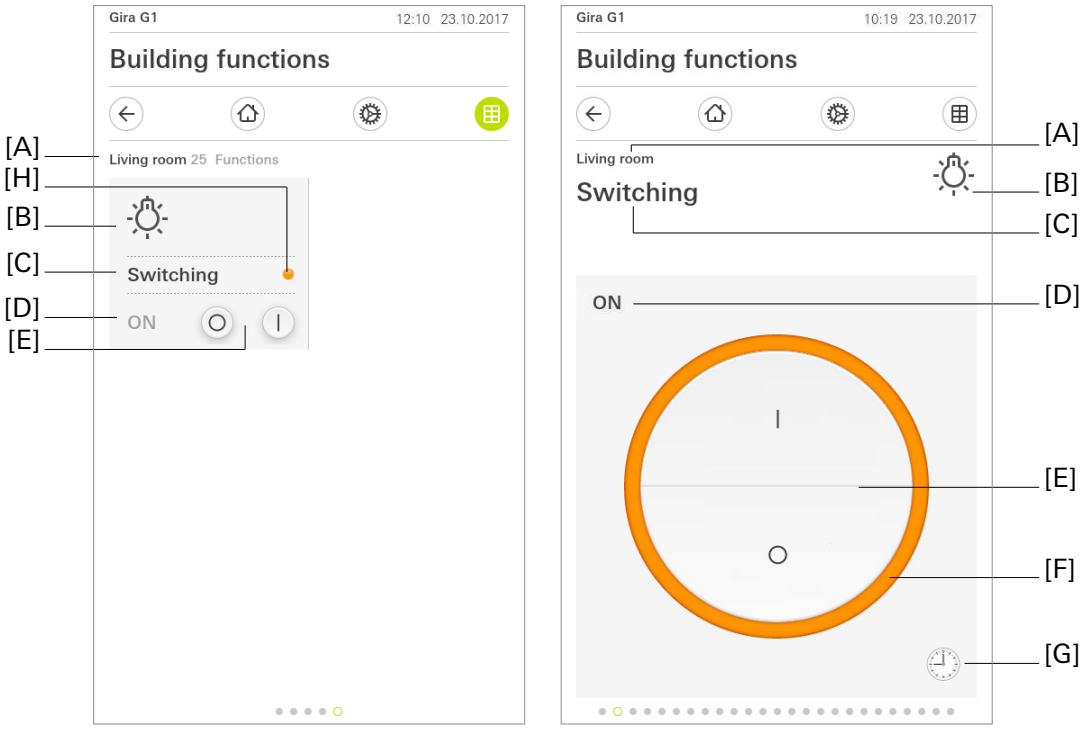


Fig 2: Example of operating and display element of the "Switching" function (rocker function).

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of "Status text"
[E] Buttons
[F] Status ring
[G] "Timer" button
[H] Status LED

Two buttons provide the rocker function in the detail view. The upper one transmits a "1" telegram and the lower one transmits a "0" telegram to the KNX. The function of the element is indicated by the corresponding icon ("I" = ON, "O" = OFF) on the rocker.
The switching function can be executed in a time-controlled manner using the "Timer" button. You have the option of switching the parameterised function on or off.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The switching status received through communication objects is indicated in the tile view by means of the parameterisable background colour of the button [E] and in the detail view by the colour of the status ring [F]. The colour can be configured separately for both switching conditions.

In addition, a status text [D] can be integrated for both views. The status text provides different information in the Gira G1 user interface (e.g. "Off" / "On" or "Absent" / "Present"), depending on the object value of the switching feedback ("0" / "1"). The text to be displayed can be parameterised. The status text is displayed as "---" for as long as no feedback object value has been received after a device reset.

A status LED can also be shown in the tile view. The colour of the status LED is defined through the parameters "Colour of the status displays for I - ON" and "Colour of the status displays for 0 - OFF".

You can choose to not use the status text and status LED. The status ring is also no longer shown in the detail view if the checkmark is deleted from the "Display status" parameter.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Switching" function has two KNX communication objects:

- "Channel n - switching" (1-bit transmitting): When you touch the button in the user interface, this object emits switching telegrams via the KNX. This allows you to directly control a switching channel of a KNX switching actuator, for example.
- "Channel n - switching feedback" (1-bit receiving):
  This object allows the device to receive a switching status feedback message, e.g. emitted by a switching actuator. Depending on the parameterisation, this influences the colour of the status displays and status text that can be shown in the display area of the channel element. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly.
  The switching status must be fed back through the "active transmitting" actuator. If the KNX switching actuator being controlled does not have a separate status feedback, the "Channel n - switching feedback" object can be connected with the "Channel n - switching" object by means of an identical group address.

### 6.2.2. Trigger ON/OFF

Using the "Trigger ON/OFF" channel function makes it possible to emit 1-bit switching telegrams via the KNX. Contrary to the "Switching" channel function, however, it is always only an ON or an OFF telegram that is transmitted, depending on the configuration. When combined with switching actuators, this makes it possible to start actions that can be interrupted or terminated, for which no further condition information can be accessed through the bus. Examples include a garage door opener or activation of a logic sequence.

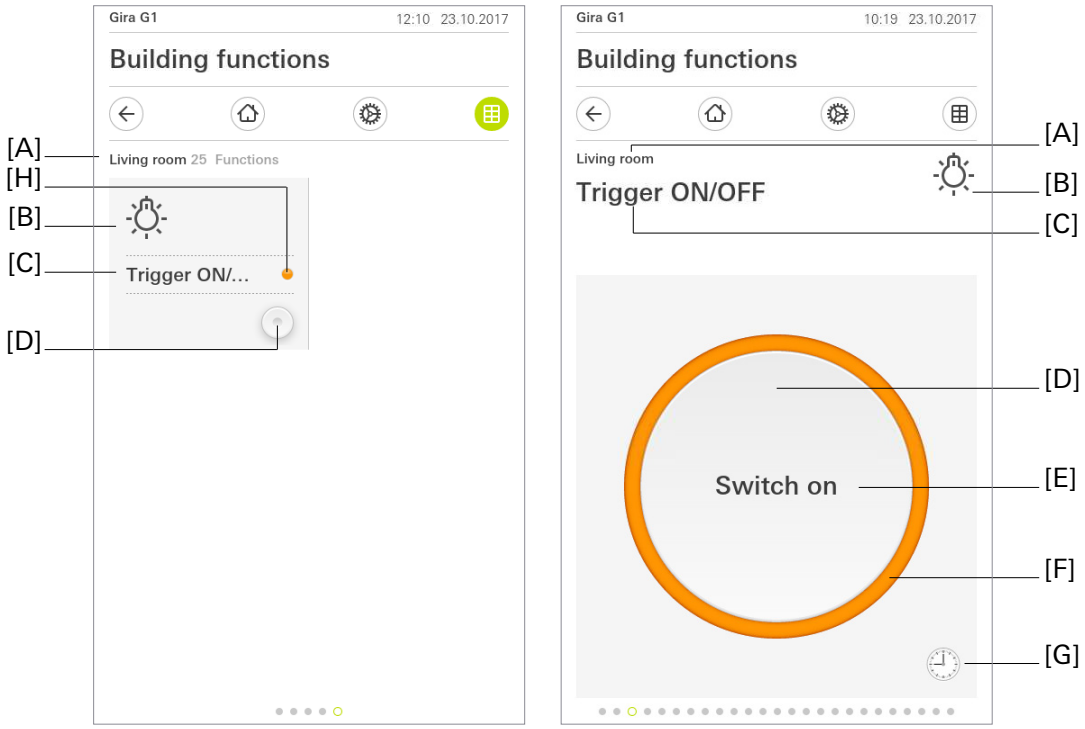


Fig 3: Example of operating and display element of the "Trigger ON/OFF" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Buttons
[E] Display of "Text"
[F] Status ring
[G] "Timer" button
[H] Status LED

The user interface is similar to a "Switching" channel function configured as a button or the "Scene extension" channel function, and contains a small button [D] in the tile view and a large button [D] in the detail view, which transmit the pre-configured ON or OFF 1-bit switching telegram to the bus when pressed.

It is only possible to transmit either an ON or an OFF switching telegram per "Trigger ON/OFF" channel function - the "Switching" or "Press / release ON/OFF" channel functions are designed for sending both telegrams.

The parameterised function can be executed in a time-controlled manner using the "Timer" button.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The user interface of the "Trigger ON/OFF" does not recognise status through a feedback object, meaning that it is always shown without colour in "idle state" and displays colour in the status ring [F] and the status LED [H] for a fixed period of time upon activation.

### Communication objects

The name of the communication object is based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Trigger ON/OFF" function has one KNX communication object:

- "Channel n - trigger" (1-bit transmitting):
  When you touch the button, this object emits 1-bit switching telegrams to the KNX.

### 6.2.3. Press / release ON/OFF

Using the "Press / release ON/OFF" channel function makes it possible to emit 1-bit switching telegrams via the KNX.

Contrary to the "Switching" channel function, however, the ON switching telegram is sent when the button is pressed, and the OFF switching telegram is sent when it is released.

This allows you to trigger brief switching actions that are precisely timed in their effect, for example when moving a room separator into an intermediate position.

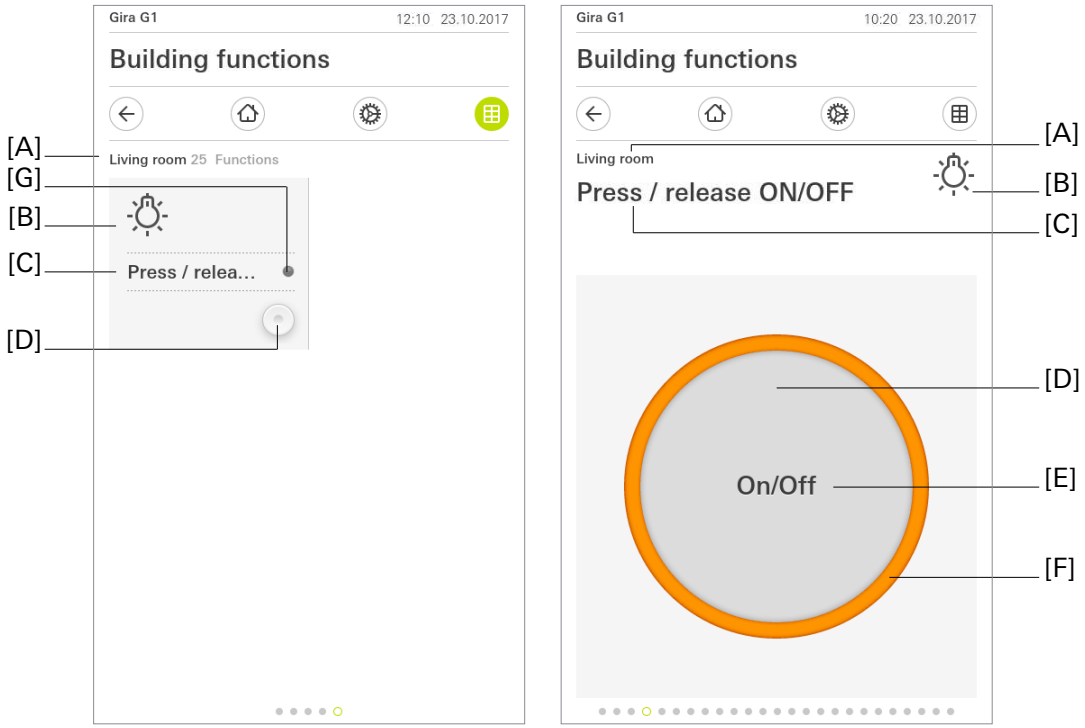


Fig 4: Example of operating and display element of the "Press / release ON/OFF" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Buttons
[E] Display of "Text"
[F] Status ring
[G] Status LED

The user interface is similar to a "Switching" channel function configured as a button or the "Scene extension" channel function, and contains a small button [D] in the tile view and a large button [D] in the detail view, which transmit the configured 1-bit ON switching telegram to the bus when touched and transmit the configured OFF telegram to the bus when released.

#### Status elements

A status text or value cannot be configured for the "Press / release ON/OFF" channel function.

The status elements [F] and [G] are displayed in the configured colour while actuated.

### Communication objects

The name of the communication object is based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Press / release ON/OFF" function has one KNX communication object:

- "Channel n - press and release" (1-bit transmitting):
  When you touch and release the button, this object emits 1-bit switching telegrams to the KNX.

### 6.2.4. Dimming brightness value

Using the "Dimming brightness value" channel function makes it possible to send out 1-byte brightness value telegrams (absolute dimming via 1-byte values as per KNX DPT 5.001 Scaling) via the KNX. When combined with dimming actuators, this makes it possible to control lighting systems. KNX speed regulators can be addressed in the same way, allowing motors to be switched on and off and modified in speed of rotation.

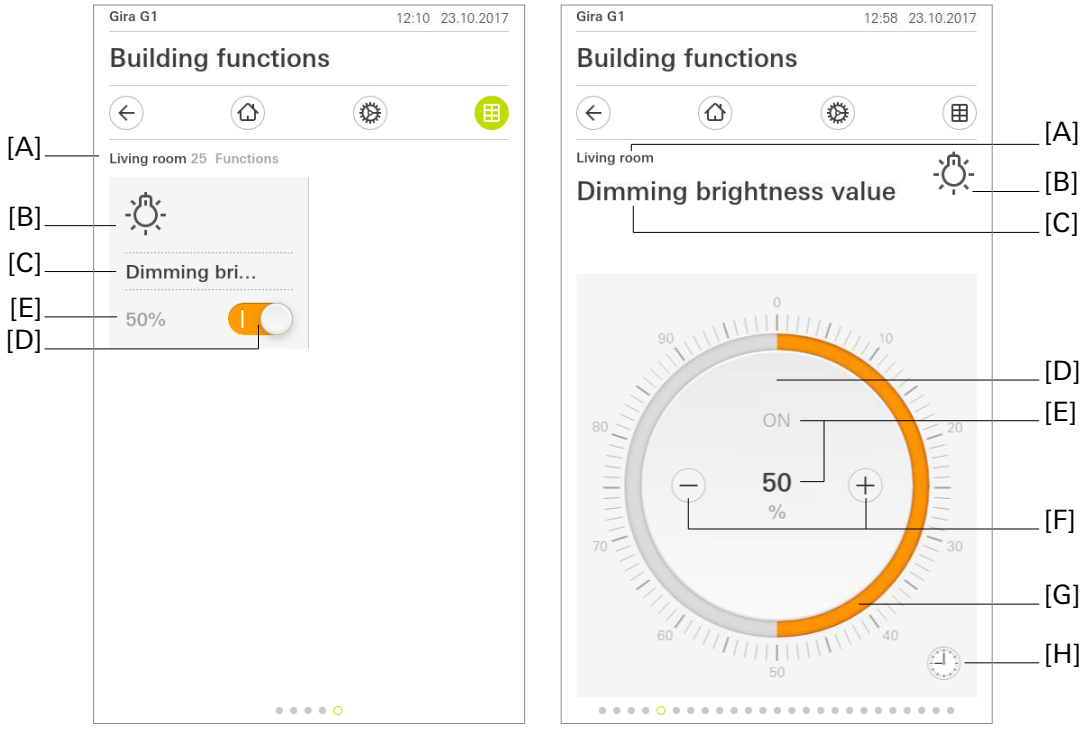


Fig 5: Example of operating and display element of the "Dimming brightness value" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Button (ON / OFF)
[E] Display of "Status value"
[F] Buttons (value adjustment)
[G] Operable scale
[H] "Timer" button

Briefly pressing the button triggers value telegrams. Depending on the button pressed, the brightness value is moved up or down by the parameterised step width. A long press of buttons ⊕ or ⊖ gradually emits brightness value telegrams to the KNX at a defined interval.
Alongside use of the buttons ⊕ or ⊖, the detail view offers the option of setting the brightness value of the lighting via an operable scale [G]. Value specifications can be triggered by continuously dragging the scale or tapping on the desired brightness value on the scale. No values are transmitted to the KNX during the dragging process. The brightness of the lighting being controlled only changes when you are finished operating the control.

It is generally recommended that you hold your finger on a point on the scale for approx. 0.5 s before beginning to use it to make your adjustment.

A switching process that switches the controlled load on and off can be implemented by tapping on the button (ON / OFF). You can only transmit switching telegrams to the KNX when operating in the tile view.

The dimming function can be executed in a time-controlled manner using the "Timer" button. The dimming values available range from 0 % to 100 % in 10 % increments.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The status value is visualised depending on the object value of the dimming value feedback from the dimming actuator being addressed. Brightness values are shown in plain text (0...100%). The status value is displayed as "-%" for as long as no feedback object value has been received after a device reset.

An operable scale [G] is displayed in the detail view. This scale shows brightness values using the dimming value feedback from the dimming actuator being addressed.

There is another way to illustrate the switching status in the tile view.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Dimming brightness value" function has four KNX communication objects:

- "Channel n - switching" (1-bit transmitting):
  When you touch the button in the user interface, this object emits switching telegrams to the KNX. This allows you to directly control a switching channel of a KNX dimming actuator, for example.
- "Channel n - feedback switching" (1-bit receiving):
  This object allows the device to receive a switching status feedback message, e.g. emitted by a dimming actuator. Depending on the parameterisation, this influences the colour of the status displays and status text that can be shown in the display area of the channel element. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The switching status must be fed back through the "active transmitting" actuator.
- "Channel n - brightness value" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control a dimming actuator.
- "Channel n - feedback brightness value" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by a dimming actuator. This influences the status value, as well as the operable scale. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.

The communication objects interpret values to be transmitted or received based on KNX data point type 5.001 (Scaling). Decimal data values 0...255 are interpreted as percentage values 0...100%.

The value of the object "Channel n - feedback brightness value" is not evaluated during operation. A long press of the buttons ⊕ or ⊖ means a continuous value adjustment, taking into account the configured step width. During the value adjustment, all status elements are continuously updated through value simulation. Only once you are finished operating the control do the display elements react to the feedback object value from the dimming actuator again.

### 6.2.5. Dimming relative

In addition to the 1-byte dimming value telegrams, using the "Dimming relative" channel function allows you to send out 4-bit dimming telegrams (relative dimming via 4 bit in accordance with DPT 3.007 Control Dimming) through the KNX. When combined with suitable actuators, this makes it possible to control lighting systems through short-term and long-term telegrams.

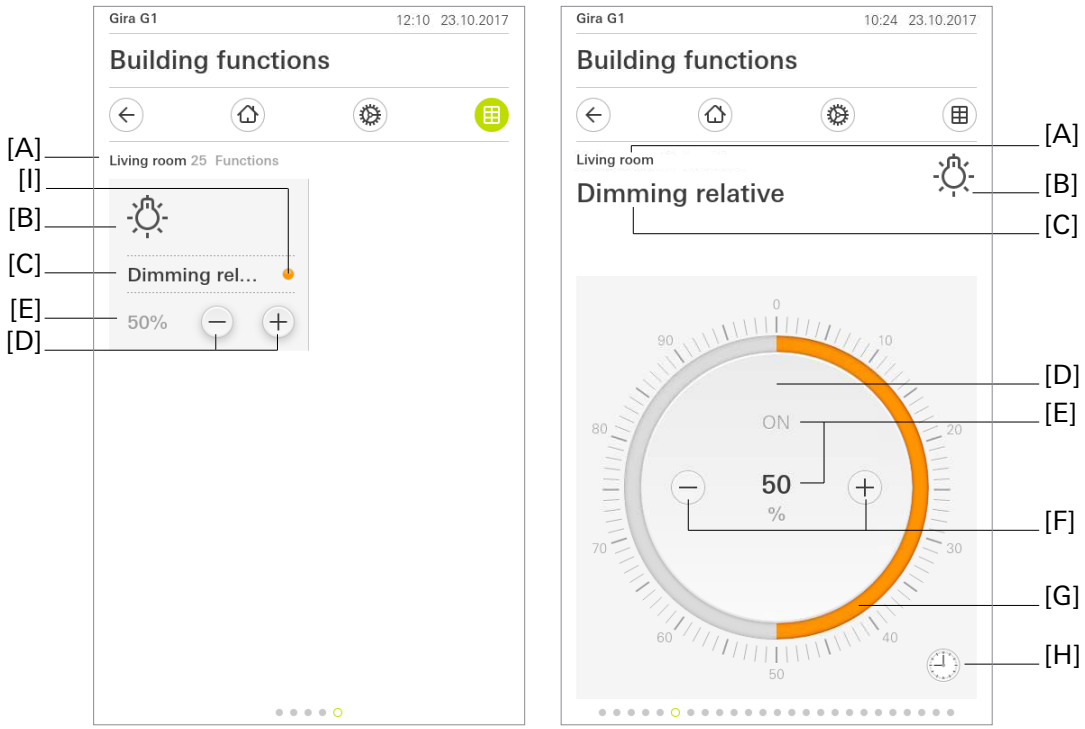


Fig 6: Example of operating and display element of the "Dimming relative" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Button (ON / OFF)
[E] Display of "Status value"
[F] Buttons (value adjustment)
[G] Operable scale
[H] "Timer" button
[I] Status LED

The buttons ⊕ and ⊖ in the tile and detail view distinguish between short and long actuations. Different telegrams are emitted to the KNX based on this.

- When you press the buttons ⊕ or ⊖, the parameterised "Time between short and long-term command" begins. If the pressed button is released again during this time, the device emits an On telegram in the case of the ⊕ button and an Off telegram in the case of the ⊖ button.
  The "Time between short and long-term command" is configured in the ETS and should be set as shorter than the short-term operation of the actuator so as not to cause flickering of lights.
- If the button is pressed for longer than the parameterised time, the Gira G1 emits a long-term telegram to dim the lights at the end of such.
- After a long press, the Gira G1 emits a telegram to stop the dimming process upon release.

The brightness value can be set using an operable scale [G] in the detail view. Value specifications can be triggered by continuously dragging the scale or tapping on the desired brightness value on the scale. No

values are transmitted to the KNX during the dragging process. The brightness of the lighting being controlled only changes when you are finished operating the control. A switching process that switches the controlled load on and off can be implemented by tapping on the button (ON / OFF).

It is generally recommended that you hold your finger on a point on the scale for approx. 0.5 s before beginning to use it to make your adjustment.

The dimming function can be executed in a time-controlled manner using the "Timer" button. The dimming values available range from 0 % to 100 % in 10 % increments.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The status value is visualised depending on the object value of the dimming value feedback from the dimming actuator being addressed. Brightness values are shown in plain text (0...100%). The status value is displayed as "-%" for as long as no feedback object value has been received after a device reset.

An operable scale [G] is shown in the display element in the detail view of the "Dimming brightness value" channel function. This scale shows brightness values in graphical format using the dimming value feedback from the dimming actuator being addressed. The scale increases clockwise as the brightness of the lighting being controlled increases. Accordingly, the scale reduces as the brightness reduces. Maximum brightness is set at the full end of the scale. If the scale does not show a range, this means that the lighting is switched off.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Dimming brightness value" function has five KNX communication objects:

- "Channel n - switching" (1-bit transmitting):
  When you touch the button in the user interface, this object emits switching telegrams to the KNX. This allows you to directly control a switching channel of a KNX dimming actuator, for example.
- "Channel n - feedback switching" (1-bit receiving):
  This object allows the device to receive a switching status feedback message, e.g. emitted by a dimming actuator. Depending on the parameterisation, this influences the colour of the status displays and status text that can be shown in the display area of the channel element. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The switching status must be fed back through the "active transmitting" actuator.
- "Channel n - brightness value" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control a dimming actuator.
- "Channel n - feedback brightness value" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by a dimming actuator. This influences the status value, as well as the operable scale. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly.
  The brightness value must be fed back through the "active transmitting" actuator.
- "Channel n - dimming" (4-bit transmitting):
  In the event of a long press of the function, this object emits dimming telegrams to the KNX. This makes it possible to control a dimming actuator.

### 6.2.6. Dimming RGB

Using the "Dimming RGB" channel function makes it possible to emit the following telegrams via the KNX:

- Switching KNX DPT 1.001
- Brightness value KNX DPT 5.001
- Red percentage KNX DPT 5.001
- Green percentage KNX DPT 5.001
- Blue percentage KNX DPT 5.001

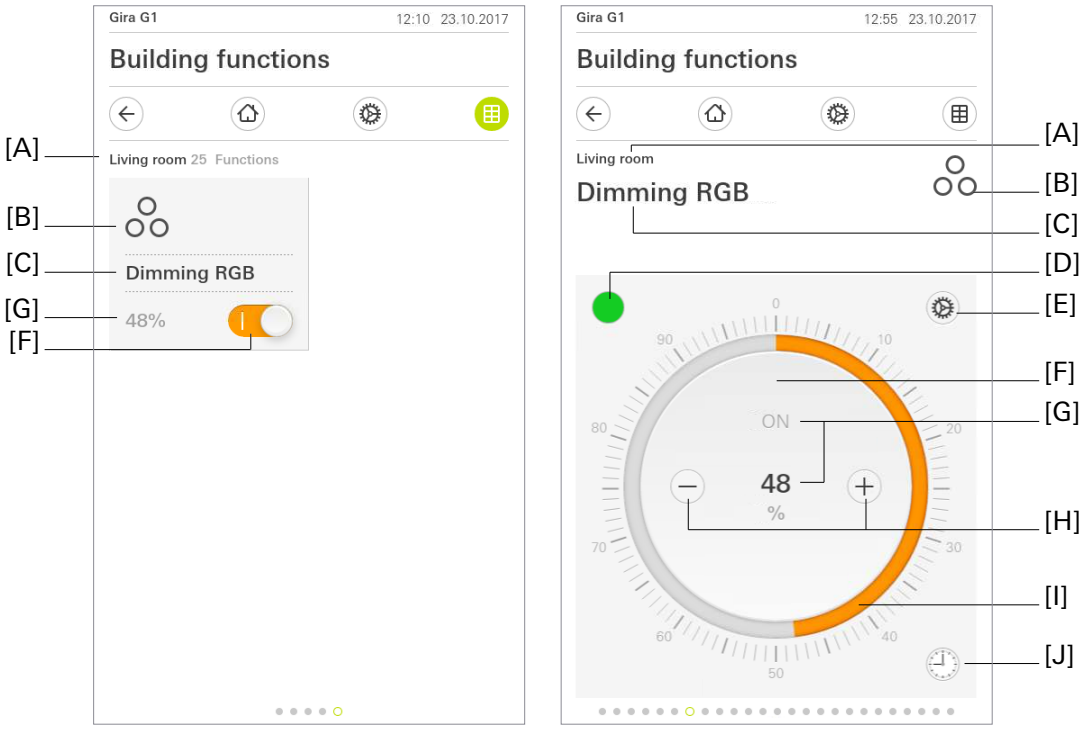


Fig 7: Example of operating and display element of the "Dimming RGB" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] "Selected colour value" button
[E] "Settings" button
[F] Button (ON / OFF)
[G] Display of "Status value"
[H] Buttons (value adjustment)
[I] Operable scale
[J] "Timer" button

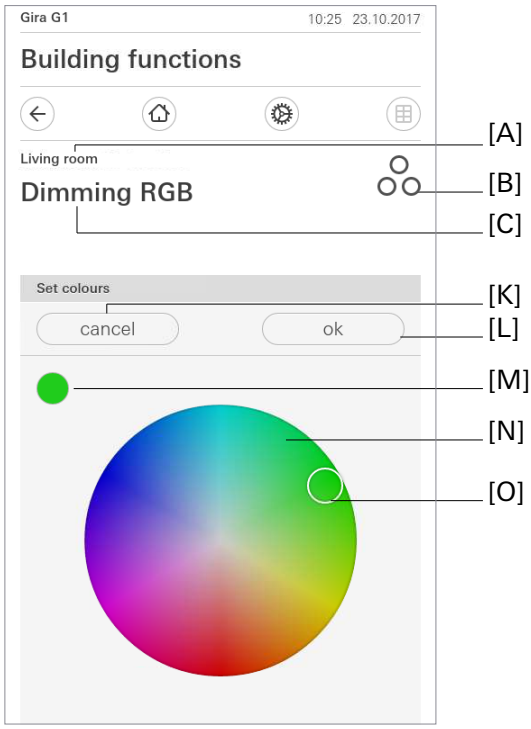


Fig 8: "Dimming RGB" - set colour

[M] Display of "Selected colour value"
[K] "Cancel" button
[L] "OK" button
[N] Operable colour wheel
[O] "Selection area" moveable element

In the tile view, the "Dimming RGB" control template is identical to the dimmer's control template. The optional status LED [H] shows the current switching condition of the channel. [G] is the selected brightness value. A switching process that switches the controlled load on and off can be implemented by tapping on the button (ON / OFF) [F]. You can only transmit switching telegrams to the KNX when operating in the tile view. Specification of a brightness value requires you to switch to the detail view.

The current brightness value of the dimming channel being controlled is visualised with an operable scale in the user interface, as well as optionally presented as a numerical value. Scaling is in percentage increments. The display reacts to the feedback telegrams from the dimming actuator in the same way as the status display. It is generally recommended that you hold your finger on a point on the scale for approx. 0.5 s before beginning to use it to make your adjustment.

The status value display [G] can be shown in both views. The status value is visualised depending on the object value of the dimming value feedback from the dimming actuator being addressed. Brightness values are shown in plain text (0...100%). The status value is displayed as "---" for as long as no feedback object value has been received after a device reset.

A short press of a button (⊕ or ⊖) triggers value telegrams. Depending on the button pressed, the brightness value is moved up or down by the parameterised step width. A long press of buttons ⊕ or ⊖ gradually emits brightness value telegrams to the KNX at a defined interval.

An operable scale [I] is shown in the display element in the detail view of the "Dimming brightness value" channel function. This scale shows brightness values in graphical format using the dimming value feedback from the dimming actuator being addressed. The scale increases clockwise as the brightness of the lighting being controlled increases. Accordingly, the scale reduces as the brightness reduces. Maximum brightness is set at the full end of the scale. If the scale does not show a range, this means that the light-

ing is switched off. The operable scale is always visible and complements the status display through status value and LED. There is another way to illustrate the switching status in the tile view. A status LED can only be shown in this view. The colour of the status LED is preset and cannot be changed. You can choose to not use the status text and status LED.

Over and above the dimmer control options, in the top left there is the "Selected colour value" button [D], which shows the colour currently selected. Clicking on it brings up an operable colour wheel [N], which allows you to select a colour. A white ring - the "Selection area" moveable element [O] - can be moved by pressing and dragging or clicking on the colour wheel [N]. If you press beyond the edge of the colour wheel, the value nearest to the edge of the colour wheel is selected and shown.

The change has an immediate effect on the light source. The values for RGB are transmitted to the actuator in individual telegrams. The feedback messages from the actuator are interpreted immediately and cause a shift in the selection area [O] and the selected colour value [D] in the event of a deviating feedback message. Any delayed feedback of individual values can mean that the position of the selection area [O] is in deviation during the period up until complete feedback is received. The "Selected colour value" button [D] shows the feedback from the actuator to transmitted values, so that a change may be delayed or displayed as a multiple toggle. "OK" [M] confirms the selection; "Cancel" [L] takes you back to the colour that was set on the page before the change. In both cases, the user is taken back to the "Set brightness" page with the operable scale [I].

**Colour space**

The selection of colour and the selection of brightness are independent of one another. The brightness is controlled through the "Set brightness" page and does not change the colour setting; selecting a colour from the colour wheel on the "Set colour" page does not change the brightness. As such, it is important that the actuator interprets the brightness from the RGB values in exactly the same way as they are calculated by the application, using the HSB colour space.

The application uses the HSB (Hue, Saturation, Brightness) colour space to issue RGB values. The brightness shown in the application always corresponds to the value of the "Brightness" object function and is independent of the values of the RGB object functions within the HSB colour space. Accordingly, the application only writes to the "Brightness" object function when the brightness is selected

When the brightness is calculated from the RGB values it is important that the actuator uses the HSB colour space to control the luminaire. If the colour space is different, it is possible that using the colour selection will cause a change in brightness and, vice versa, that changing the brightness will cause deviations in colour.

The "Settings" button takes you to the context menu. Here you can save the current colour and brightness value setting as a default using the "Save current setting" button. Up to five default values can be saved under "Stored settings". Each of these is displayed in a row with the colour and percentage value. When you select a default value, the brightness is first written to the "Brightness" object function. Directly after, the three RGB values are written to the relevant object functions in succession. All default values currently present can also be called up in a time-controlled manner using the function timer.

The function can be executed in a time-controlled manner using the "Timer" button. The options are "switch on" and "switch off", as well as all colour and brightness settings stored at the point of setting. When you trigger a default value, a "1" is first written to the switching object function. Directly after, the individual values corresponding to the default value are written to the brightness and RGB object functions.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Dimming RGB" function has ten KNX communication objects:

- "Channel n - switching" (1-bit transmitting):
  When you touch the button in the user interface, this object emits switching telegrams to the KNX. This allows you to directly control a switching channel of a KNX dimming actuator, for example.
- "Channel n - feedback switching" (1-bit receiving):
  This object allows the device to receive a switching status feedback message, e.g. emitted by an actuator. Depending on the parameterisation, this influences the colour of the status displays and status text that can be shown in the display area of the channel element. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The switching status must be fed back through the "active transmitting" actuator.
- "Channel n - brightness value" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - feedback brightness value" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the operable scale. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.
- "Channel n - red percentage" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - red percentage" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the selection area in the operable colour wheel. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.
- "Channel n - green percentage" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - feedback green percentage" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the selection area in the operable colour wheel. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.
- "Channel n - blue percentage" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.

- "Channel n - feedback blue percentage" (1-byte receiving):
  This optional object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the selection area in the operable colour wheel. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.

The communication objects interpret values to be transmitted or received based on KNX data point type 5.001 (Scaling). Decimal data values 0...255 are interpreted as percentage values 0...100%.

The value of the object "Channel n - brightness value feedback" is not evaluated during operation. A long press of the buttons ⊕ or ⊖ means a continuous value adjustment, taking into account the configured step width. As part of this value adjustment, all status elements of the channel element are continuously updated through value simulation. Only once you are finished operating the control do the display elements react to the feedback object value from the dimming actuator again.

### 6.2.7. Dimming RGBW

Using the "Dimming RGBW" channel function makes it possible to emit the following telegrams via the KNX:

- Switching KNX DPT 1.001
- Brightness value KNX DPT 5.001
- Red percentage KNX DPT 5.001
- Green percentage KNX DPT 5.001
- Blue percentage KNX DPT 5.001
- Blue percentage KNX DPT 5.001
- White percentage KNX DPT 5.001

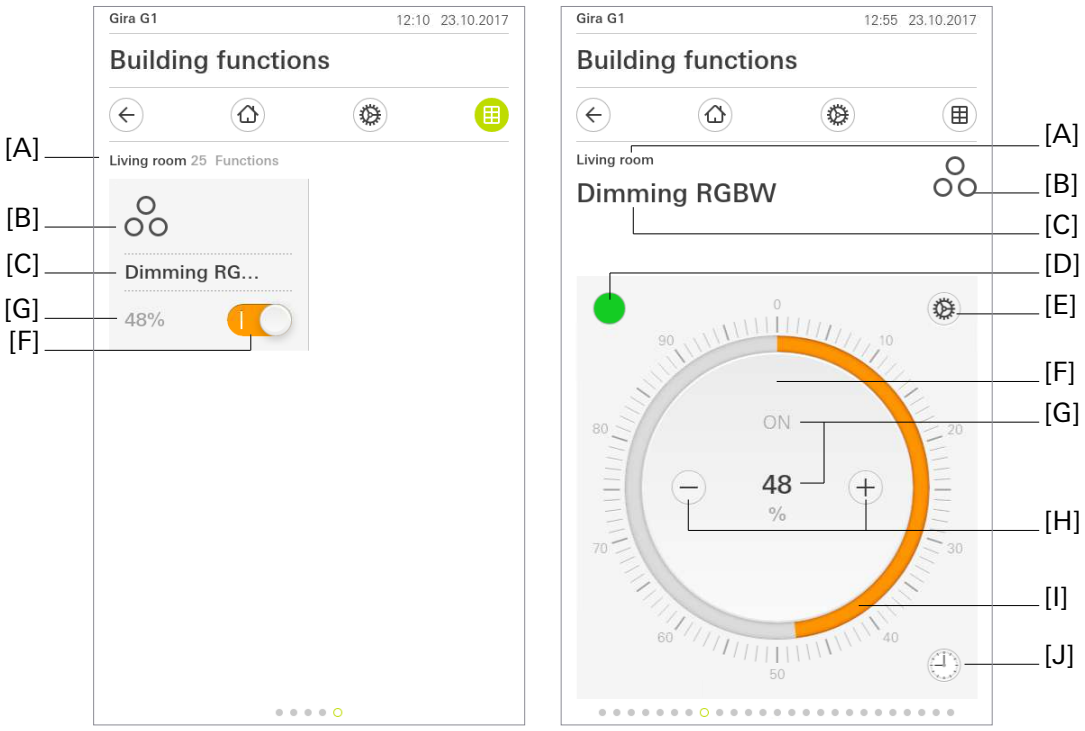


Fig 9: Example of operating and display element of the "Dimming RGBW" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] "Selected colour value" button
[E] "Settings" button
[F] Button (ON / OFF)
[G] Display of "Status value"
[H] Buttons (value adjustment)
[I] Operable scale
[J] "Timer" button

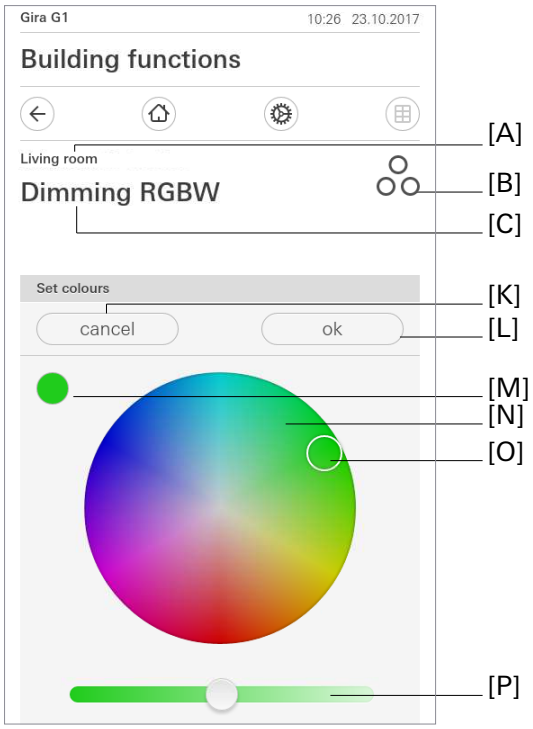


Fig 10: "Dimming RGBW" - set colour

[K] "Cancel" button
[L] "OK" button
[M] Display of "Selected colour value"
[N] Operable colour wheel
[O] "Selection area" moveable element
[P] "White percentage" slider

In the tile view, the "Dimming RGBW" control template is identical to the dimmer's control template. The optional status LED [H] shows the current switching condition of the channel. [G] is the selected brightness value. A switching process that switches the controlled load on and off can be implemented by tapping on the button (ON / OFF) [F]. You can only transmit switching telegrams to the KNX when operating in the tile view. Specification of a brightness value requires you to switch to the detail view.

The current brightness value of the dimming channel being controlled is visualised with an operable scale in the user interface, as well as optionally presented as a numerical value. Scaling is in percentage increments. The display reacts to the feedback telegrams from the dimming actuator in the same way as the status display. It is generally recommended that you hold your finger on a point on the scale for approx. 0.5 s before beginning to use it to make your adjustment.

The status value display [G] can be shown in both views. The status value is visualised depending on the object value of the dimming value feedback from the dimming actuator being addressed. Brightness values are shown in plain text (0...100%). The status value is displayed as "---" for as long as no feedback object value has been received after a device reset.

A short press of a button (⊕ or ⊖) triggers value telegrams. Depending on the button pressed, the brightness value is moved up or down by the parameterised step width. A long press of buttons ⊕ or ⊖ gradually emits brightness value telegrams to the KNX at a defined interval.

An operable scale [I] is shown in the display element in the detail view of the "Dimming brightness value" channel function. This scale shows brightness values in graphical format using the dimming value feedback from the dimming actuator being addressed. The scale increases clockwise as the brightness of the lighting being controlled increases. Accordingly, the scale reduces as the brightness reduces. Maximum

brightness is set at the full end of the scale. If the scale does not show a range, this means that the lighting is switched off. The operable scale is always visible and complements the status display through status value and LED. There is another way to illustrate the switching status in the tile view. A status LED can only be shown in this view. The colour of the status LED is preset and cannot be changed. You can choose to not use the status text and status LED.

Over and above the dimmer control options, in the top left there is the "Selected colour value" button [D], which shows the colour currently selected. Clicking on it brings up an operable colour wheel [N], which allows you to select a colour. A white ring - the "Selection area" moveable element [O] - can be moved by pressing and dragging or clicking on the colour wheel [N]. If you press beyond the edge of the colour wheel, the value nearest to the edge of the colour wheel is selected and shown. The white percentage of the RGBW light is controlled by the "White percentage" slider [P]. As with the three colour channels, the value is transmitted to the KNX bus upon release. A feedback message is received and displayed in both the "White percentage" slider and the "Selected colour value" area.

The change has an immediate effect on the light source. The values for RGBW are transmitted to the actuator in individual telegrams. The feedback messages from the actuator are interpreted immediately and cause a shift in the selection area [O] and the selected colour value [D] in the event of a deviating feedback message. Any delayed feedback of individual values can mean that the position of the selection area [O] is in deviation during the period up until complete feedback is received. The "Selected colour value" button [D] shows the feedback from the actuator to transmitted values, so that a change may be delayed or displayed as a multiple toggle. "OK" [L] confirms the selection; "Cancel" [M] takes you back to the colour that was set on the page before the change. In both cases, the user is taken back to the "Set brightness" page with the operable scale [I].

**Colour space**

The selection of colour and the selection of brightness are independent of one another. The brightness is controlled through the "Set brightness" page and does not change the colour setting; selecting a colour from the colour wheel on the "Set colour" page does not change the brightness. As such, it is important that the actuator interprets the brightness from the RGBW values in exactly the same way as they are calculated by the application, using the HSB colour space.

The application uses the HSB (Hue, Saturation, Brightness) colour space to issue RGBW values. The brightness shown in the application always corresponds to the value of the "Brightness" object function and is independent of the values of the RGBW object functions within the HSB colour space. Accordingly, the application only writes to the "Brightness" object function when the brightness is selected.

When the brightness is calculated from the RGBW values it is important that the actuator uses the HSB colour space to control the luminaire. If the colour space is different, it is possible that using the colour selection will cause a change in brightness and, vice versa, that changing the brightness will cause deviations in colour.

The "Settings" button takes you to the context menu. Here you can save the current colour and brightness value setting as a default using the "Save current setting" button. Up to five default values can be saved under "Stored settings". Each of these is displayed in a row with the colour and percentage value. When you select a default value, the brightness is first written to the "Brightness" object function. Directly after, the three RGBW values are written to the relevant object functions in succession. All default values currently present can also be called up in a time-controlled manner using the function timer.

The function can be executed in a time-controlled manner using the "Timer" button. The options are "switch on" and "switch off", as well as all colour and brightness settings stored at the point of setting. When you trigger a default value, a "1" is first written to the switching object function. Directly after, the individual values corresponding to the default value are written to the brightness and RGBW object functions.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Dimming RGBW" function has twelve KNX communication objects:

- "Channel n - switching" (1-bit transmitting):
  When you touch the button in the user interface, this object emits switching telegrams to the KNX. This allows you to directly control a switching channel of a KNX dimming actuator, for example.
- "Channel n - feedback switching" (1-bit receiving):
  This object allows the device to receive a switching status feedback message, e.g. emitted by an actuator. Depending on the parameterisation, this influences the colour of the status displays and status text that can be shown in the display area of the channel element. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The switching status must be fed back through the "active transmitting" actuator.
- "Channel n - brightness value" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - feedback brightness value" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the operable scale. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.
- "Channel n - red percentage" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - red percentage" (1-byte receiving):
  This optional object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the selection area in the operable colour wheel. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.
- "Channel n - green percentage" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - feedback green percentage" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the selection area in the operable colour wheel. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.
- "Channel n - blue percentage" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - feedback blue percentage" (1-byte receiving):
  This object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the selection area in the operable colour wheel. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.

- "Channel n - white percentage" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control an actuator.
- "Channel n - feedback white percentage" (1-byte receiving):
  This optional object allows the device to receive a brightness value feedback message, e.g. emitted by an actuator. This influences the status value, as well as the selection area in the operable colour wheel. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.

The communication objects interpret values to be transmitted or received based on KNX data point type 5.001 (Scaling). Decimal data values 0...255 are interpreted as percentage values 0...100%.

The value of the object "Channel n - brightness value feedback" is not evaluated during operation. A long press of the buttons ⊕ or ⊖ means a continuous value adjustment, taking into account the configured step width. As part of this value adjustment, all status elements of the channel element are continuously updated through value simulation. Only once you are finished operating the control do the display elements react to the feedback object value from the dimming actuator again.

### 6.2.8. Dimming Tunable White

As with "Dimming brightness value", using the "Dimming Tunable White" channel function makes it possible to send out 1-byte brightness value telegrams (absolute dimming via 1-byte values as per KNX DPT 5.001 Scaling) via the KNX. A colour temperature can be sent in addition to brightness.

When combined with suitable dimming actuators, this makes it possible to control lighting systems and adjust the colour temperature.

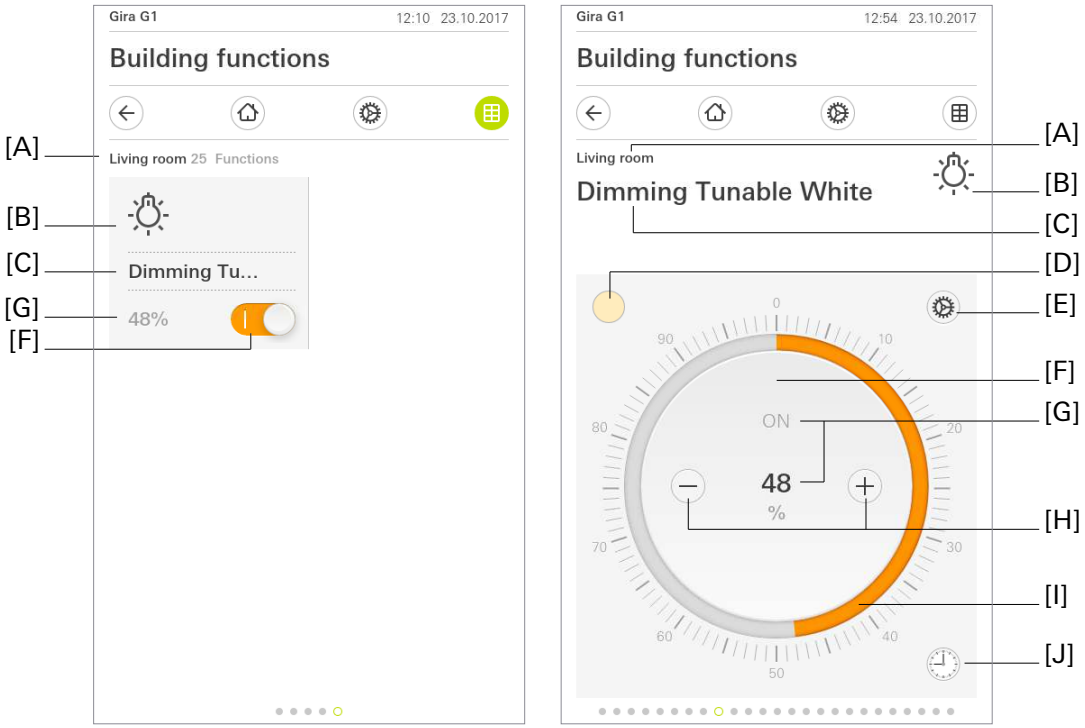


Fig 11: Example of operating and display element of the "Dimming Tuneable White" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] "Colour temperature" button
[E] "Settings" button
[F] Button (ON / OFF)
[G] Display of "Status value"
[H] Buttons (value adjustment)
[I] Operable scale
[J] "Timer" button

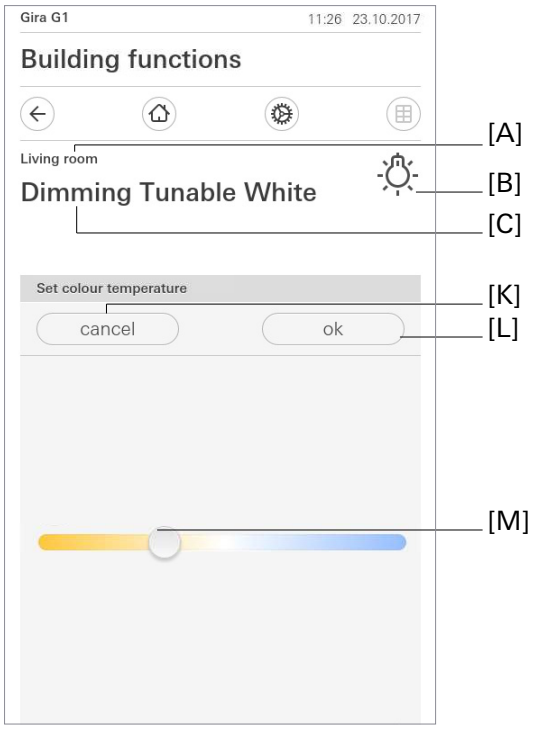


Fig 12: "Dimming Tuneable White" - set colour temperature

[K] "Cancel" button
[L] "OK" button
[M] "Colour temperature" slider

In the tile view, the "Dimming Tuneable White" control template is identical to the "Dimming brightness value" control template. [G] is the selected brightness value. A switching process that switches the controlled load on and off can be implemented by tapping on the button (ON / OFF) [F].
You can only transmit switching telegrams to the KNX when operating in the tile view. Specification of a brightness value requires you to switch to the detail view.

The current brightness value of the dimming channel being controlled is visualised with an operable scale in the user interface, as well as optionally presented as a numerical value. Scaling is in percentage increments. The display reacts to the feedback telegrams from the dimming actuator in the same way as the status display.
It is generally recommended that you hold your finger on a point on the scale for approx. 0.5 s before beginning to use it to make your adjustment.

The status value display [G] can be shown in both views. The status value is visualised depending on the object value of the dimming value feedback from the dimming actuator being addressed. Brightness values are shown in plain text (0...100%). The status value is displayed as "-" for as long as no feedback object value has been received after a device reset.
A short press of a button (⊕ or ⊖) triggers value telegrams. Depending on the button pressed, the brightness value is moved up or down by the parameterised step width. A long press of buttons ⊕ or ⊖ gradually emits brightness value telegrams to the KNX at a defined interval.

An operable scale [I] is shown in the display element in the detail view of the "Dimming brightness value" channel function. This scale shows brightness values in graphical format using the dimming value feedback from the dimming actuator being addressed. The scale increases clockwise as the brightness of the lighting being controlled increases. Accordingly, the scale reduces as the brightness reduces. Maximum brightness is set at the full end of the scale. If the scale does not show a range, this means that the lighting is switched off. The operable scale is always visible and complements the display of status value. There is another way to illustrate the switching status in the tile view.

Over and above the dimmer control options, in the top left there is the "Colour temperature" button [D], which shows the colour temperature currently selected. Clicking on it brings up an operable scale for selecting the colour temperature. The scale has the "Minimum colour temperature" at the left ("cold") end and the "Maximum colour temperature" at the right ("warm") end. The selected position on the scale is linearly converted to a value between "Minimum colour temperature" and "Maximum colour temperature" and is sent to the bus as a 2-byte telegram of data type 9.002 temperature difference (K). The colour/temperature shown does not have to match the actual light from the luminaire, and is only intended to serve as a visual guide.

The change has an immediate effect on the light source. Clicking on "OK" confirms the selection; "Cancel" takes you back to the colour temperature that was set on the page before the change. In both cases, the user is taken back to the "Set brightness" page with the operable scale [I].

The "Settings" button [E] takes you to the context menu. Here you can save the current colour temperature and brightness value setting as a default using the "Save current setting" button. Up to five default values can be saved under "Stored settings". Each of these is displayed in a row with the colour temperature and percentage value. When you select a default value, the brightness is first written to the "Brightness" object function. Directly after, the colour temperature in kelvin is written to the "Colour temperature" object functions. All default values currently present can also be called up in a time-controlled manner using the timer.

The function can be executed in a time-controlled manner using the "Timer" button. The options are "switch on" and "switch off", as well as all colour and brightness settings stored at the point of setting. When you trigger a default value, a "1" is first written to the switching object function. Directly after, the colour temperature default value is written to the "Colour temperature" object function.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Dimming Tuneable White" function has six KNX communication objects:

- "Channel n - switching" (1-bit transmitting): When you touch the button in the user interface, this object emits switching telegrams to the KNX. This allows you to directly control a switching channel of a KNX dimming actuator, for example.
- "Channel n - feedback switching" (1-bit receiving):
  This optional object allows the device to receive a switching status feedback message, e.g. emitted by a dimming actuator. Depending on the parameterisation, this influences the colour of the status displays and status text that can be shown in the display area of the channel element. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly.
  The switching status must be fed back through the "active transmitting" actuator.
- "Channel n - brightness value" (1-byte transmitting):
  When the function is actuated, this object emits value telegrams (0...100%) to the KNX. This makes it possible to control a dimming actuator.
- "Channel n - feedback brightness value" (1-byte receiving):
  This optional object allows the device to receive a brightness value feedback message, e.g. emitted by a dimming actuator. This influences the status value, as well as the operable scale. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The brightness value must be fed back through the "active transmitting" actuator.

The communication objects interpret values to be transmitted or received based on KNX data point type 5.001 (Scaling). Decimal data values 0...255 are interpreted as percentage values 0...100%.

The value of the object "Channel n - dimming brightness value feedback" is not evaluated during operation. A long press of the buttons (+) and (-) means a continuous value adjustment. As part of this value adjustment, all status elements of the channel element are continuously updated through value simulation. Only once you are finished operating the control do the display elements react to the feedback object value from the dimming actuator again.

- "Channel n - colour temperature" (2-byte transmitting): When a colour temperature is activated, this object sends out the selected value in kelvin to the KNX in a format appropriate for data point type 9.002 temperature difference (K). This makes it possible to control a dimming actuator with colour temperature control.
- "Channel n - colour temperature feedback" (2-byte transmitting): This object allows the device to receive a colour temperature feedback message, e.g. emitted by a dimming actuator with colour temperature control. This influences the status display [D], as well as the operable scale for colour temperature. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly. The colour temperature must be fed back through the "active transmitting" actuator.

### 6.2.9. Blind / shutter step move

Using the "Blind / shutter step move" channel function makes it possible to send out 1-bit switching telegrams as per KNX data point types 1.007 (Step) and 1.008 (UpDown) via the KNX. When combined with suitable actuators, this makes it possible to control blinds (incl. slats) and shutters through short-term and long-term telegrams. In the same way, individual versions of the status displays make it possible to control other shading systems, such as vertical slats and awnings.

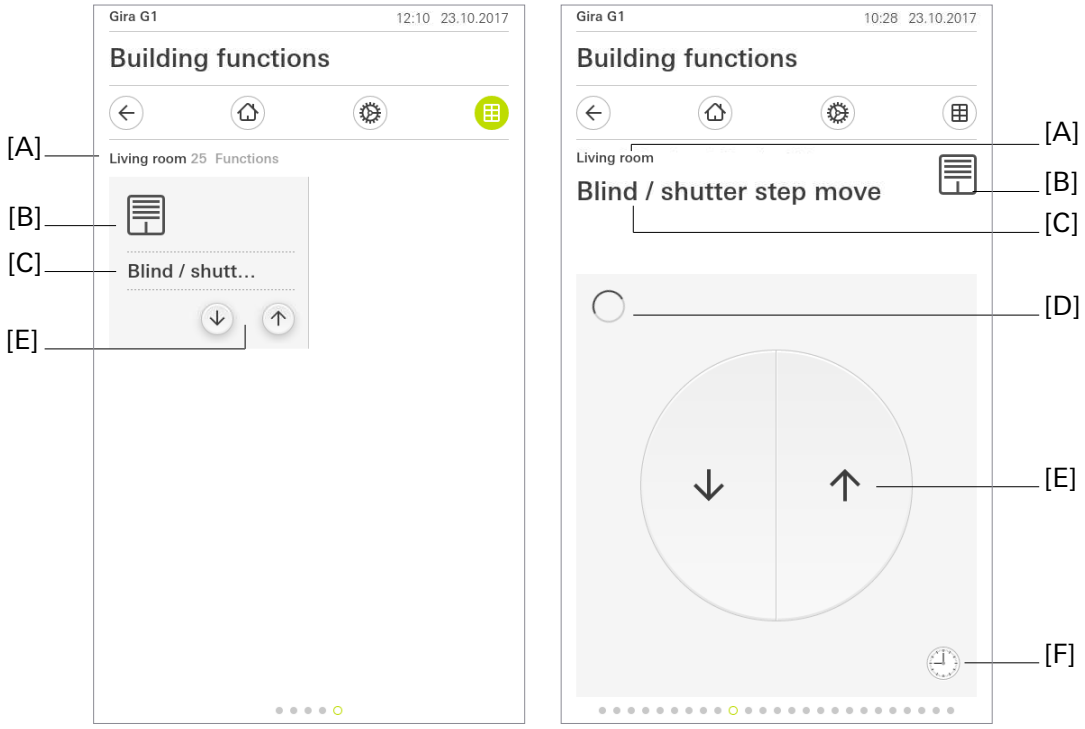


Fig 13: Example of operating and display element of the "Blind / shutter step move" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of "Drive movement"
[E] Button (UP / DOWN)
[F] "Timer" button

The current position of the selected element (slat or curtain) is not shown in the user interface.

The operating concept of this channel function is currently fixed at "Step / Move".
The button distinguishes between short and long actuations. Different telegrams are emitted to the KNX based on this:

- As soon as you tap on a button, the Gira G1 emits a short-term telegram to the KNX, which stops a travelling drive and internally starts the "Time between short and long-term command" T1. If the button is released again within T1, the device does not emit any other telegrams. This step serves to stop continuous travel.
  The "Time between short and long-term command" is configured in the ETS and should be set as shorter than the short-term operation of the actuator so as not to cause jerking of the drive being controlled.
- If the button is pressed for longer than T1, the Gira G1 emits a long-term telegram to move the drive at the end of T1.

The function can be executed in a time-controlled manner using the "Timer" button. The options are "raising" and "lowering" the curtain being controlled.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Blind / shutter step move" function has three KNX communication objects:

- "Channel n - curtain short-term operation (stop)" (1-bit transmitting): When you touch the button briefly, this object emits short-term telegrams to the KNX. This stops ongoing drive configurations. Alternatively, curtains or slats are moved for a brief period if the drive is actuated from "Stop" state. KNX blind or shutter actuators then execute the configured short-term operation (step).
- "Channel n - curtain long-term operation" (1-bit transmitting): When you touch the button for an extended time, this object emits long-term telegrams (up, down) to the KNX. This means that controlled curtains of shading systems can be moved to the end positions. KNX blind or shutter actuators execute the configured long-term operation (move). A short-term telegram (see "Channel n - curtain short-term operation (stop)" object (1-bit transmitting)) can interrupt ongoing drive configurations of long-term operation (stop), making it possible to halt a curtain at any desired point.
- "Channel n - feedback drive movement" (1-bit receiving):
  Through this object, the actuator is able to tell the device that the controlled drive (e.g. a blind) is in motion. Drive movement must be fed back through the "active transmitting" actuator.

### 6.2.10. Blind positioning

The "Blind positioning" channel function makes it possible to send out 1-byte position telegrams (values as per KNX DPT 5.001 Scaling) through the KNX. When combined with blind actuators, this makes it possible to control blinds (incl. slats) and shutters through position values. In the same way, individual versions of the status displays make it possible to control other shading systems, such as vertical slats and awnings.

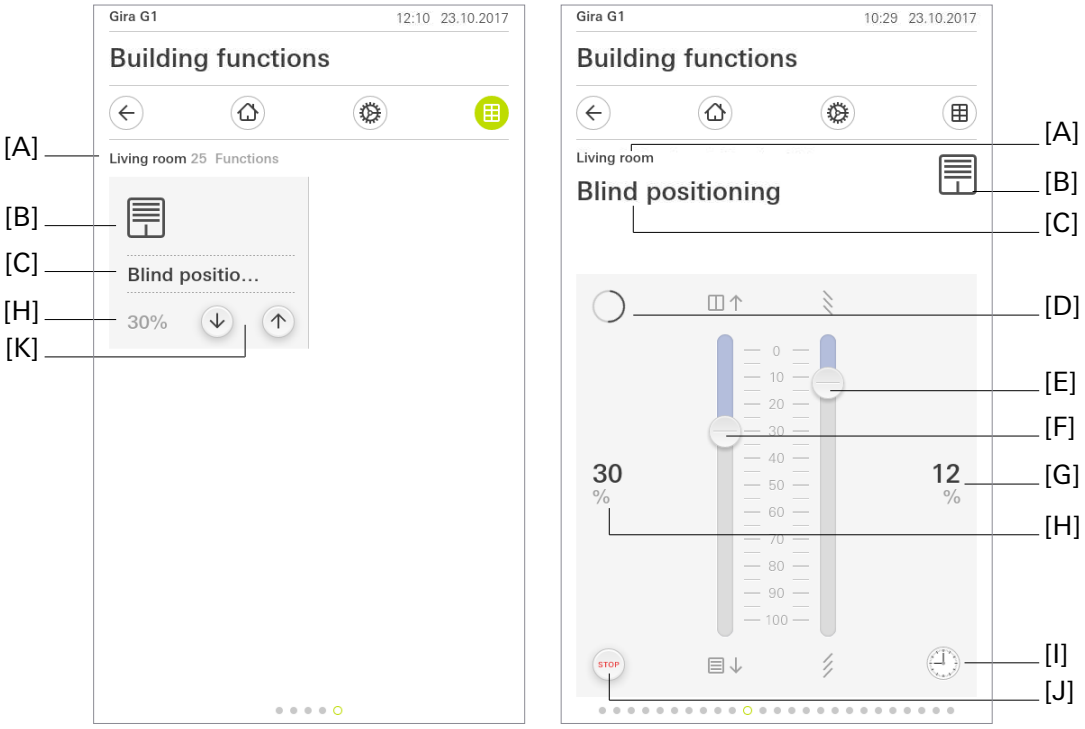


Fig 14: Example of operating and display element of the "Blind positioning" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of "Drive movement"
[E] "Slat position" slider
[F] "Curtain position" slider
[H] Display of "Status value" (curtain position)
[G] Display of "Status value" (slat position)
[I] "Timer" button
[J] "STOP" button
[K] "Curtain position" buttons

Telegrams that trigger a change of position in the curtain are generated through two buttons [D] in the tile view. Tapping on the "Curtain position down" button transmits a command to the KNX, causing the curtain to be lowered. The curtain is raised with the "Curtain position up" button. Every tap on one of these buttons causes a change of position by the configured step width. A long press of one of these buttons triggers lowering or raising to 100% or 0%.

The position of the blind and slats can be altered in the detail view. There are sliders available for both blind position changes and slat adjustment. The slider can be continuously dragged between 0 and 100 per cent using your finger. The desired position values are set by dragging the slider. Once the slider has been pushed to the desired position, the precise percentage value is transmitted to the KNX through the appropriate communication object.

In the detail view, active drive movement is fed back through a rotating drive icon. A prerequisite for this feedback is for the blind actuator being controlled to actively be able to feed back the drive movements to the KNX.

The user can stop active travel on the Gira G1 at any time. Active movement of the curtain or adjustment of slats is stopped by pressing the button ⓐ or ⓑ briefly in the tile view and pressing the STOP button in the detail view. The movement is stopped directly using buttons [D] and [J]. The curtain then stops immediately at its current position.

A slider for constant value adjustment is not available in the tile view! Here it is possible to adjust values by the step width configured in the ETS by means of a short or long press of the buttons "Curtain position down" and "Curtain position up".

The function can be executed in a time-controlled manner using the "Timer" button. The curtain heights available range from 0 % to 100 % in 10 % increments.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

**Status elements**

The status values for curtain position [E] and for slat position [G] are shown in the Gira G1. The status value for the curtain position [E] is received by the "Feedback curtain position" communication object. The status value for the slat position [G] is received by the "Feedback slat position" communication object. It can only be read within the detail view. The status value is visualised depending on the object value of the curtain position or slat position feedback from the blind actuator being addressed. Position values are shown in plain text (0...100%). The status value is displayed as "-%" for as long as no feedback object value has been received after a device reset.

Two sliders [F] and [H] are shown in the display element in the detail view of the "Blind positioning" channel function. These sliders show position values in graphical format using the position value feedback from the blind actuator being addressed. The position of the sliders increases as the curtain is raised or the slat position is opened. Similarly, the sliders [F] and [H] decrease to up to 100% as the curtain lowers or the slats are closed. The sliders [F] and [H] are always visible in the detail view and complement the "Status value" display.

The function icon for this channel function can be configured in the ETS. The display icons shown at the top and bottom end of the sliders [F] and [H] are fixed and cannot be changed. They show the user the reaction that is to be expected when the slider is moved in the corresponding direction.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Blind positioning" function has seven KNX communication objects:

- "Channel n - curtain short-term operation (stop)" (1-bit transmitting):
  When you press the "STOP" button in the detail view or briefly tap the "Curtain position" button in the tile view, this object stops a moving curtain or a slat in its current position.
- "Channel n - curtain long-term operation" (1-bit transmitting): When you touch the button for an extended time, this object emits long-term telegrams (up, down) to the KNX. This means that controlled curtains of shading systems can be moved to the end positions. KNX blind or shutter actuators execute the configured long-term operation (move). A short-term telegram (see "Channel n - curtain short-term operation (stop)" object (1-bit transmitting)) can interrupt ongoing drive configurations of long-term operation (stop), making it possible to halt a curtain at any desired point.
- "Channel n - feedback drive movement" (1-bit receiving): Through this object, the actuator is able to tell the device that the controlled drive (e.g. a blind) is in motion.
  The position value must be fed back through the "active transmitting" actuator.
- "Channel n - curtain position" (1-byte transmitting):
  When you press the buttons or operate the slider, this object emits value telegrams (0...100%) for curtain height specification to the KNX.
- "Channel n - feedback curtain position" (1-byte receiving): This object allows the device to receive a position feedback message regarding curtain height, e.g. emitted by a blind actuator. This influences the status value and slider for curtain height visualisation. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly.
  The position value must be fed back through the "active transmitting" actuator.
- "Channel n - slat position" (1-byte transmitting):
  When the buttons are pressed, this object emits value telegrams (0...100%) for slat position specification to the KNX.
- "Channel n - feedback slat position" (1-byte receiving): This object allows the device to receive a position feedback message regarding slat position, e.g. emitted by a blind actuator. This influences the status value and slider for slat visualisation. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly.
  Once again here, the position value must be fed back through the "active transmitting" actuator.

The communication objects interpret values to be transmitted or received based on KNX data point type 5.001 (Scaling). Decimal data values 0...255 are interpreted as percentage values 0...100%.

The value of the objects "Channel n - feedback curtain position" and "Channel n - feedback slat position" is not evaluated during operation. Only once you are finished operating the control do the display elements react to the feedback object value from the blind actuator again.

### 6.2.11. Shutter positioning

The "Shutter positioning" channel function makes it possible to send out 1-byte position telegrams (values as per KNX DPT 5.001 Scaling) through the KNX. When combined with shutter actuators, this makes it possible to control shutters through position values. In the same way, individual versions of the status displays make it possible to control other shading systems, such as awnings.

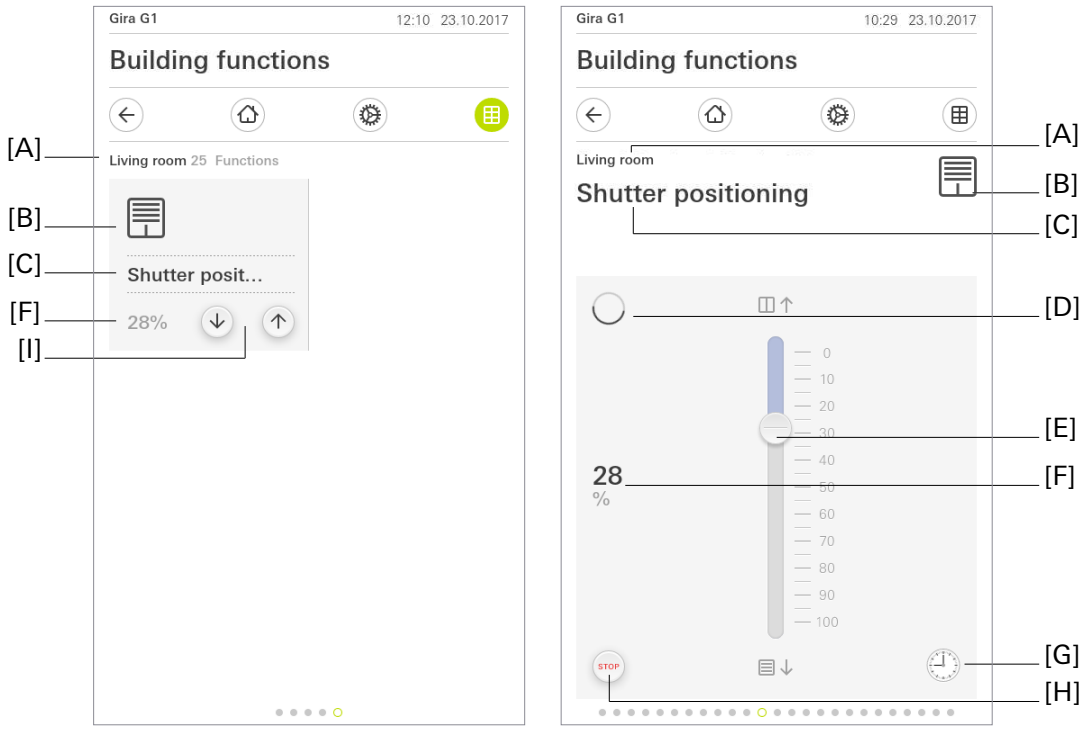


Fig 15: Example of operating and display element of the "Shutter position" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of "Drive movement"
[E] "Curtain position" slider
[F] Display of "Status value" (curtain position)
[G] "Timer" button
[H] "STOP" button
[I] "Curtain position" buttons

Telegrams that trigger a change of position in the curtain are generated through two buttons [I] in the tile view. Tapping on the "Curtain position down" button transmits a command to the KNX, causing the curtain to be lowered. The curtain is raised with the "Curtain position up" button. Every tap on one of these buttons causes a change of position by the configured step width. A long press of one of these buttons triggers lowering or raising to 100% or 0%.

The position of the shutter can be changed in the detail view. A slider is available to change the position of the shutter. The slider can be continuously dragged between 0 and 100 per cent using your finger. The desired position values are set by dragging the slider. Once the slider has been pushed to the desired position, the precise percentage value is transmitted to the KNX through the appropriate communication object.

In the detail view, active drive movement is fed back through a rotating drive icon. A prerequisite for this feedback is for the shutter actuator being controlled to actively be able to feed back the drive movements to the KNX.

The user can stop active travel on the Gira G1 at any time. Active movement of the curtain is stopped by pressing the button ⓐ or ⓑ briefly in the tile view and pressing the ⓢ button in the detail view. The curtain then stops immediately at its current position.

A slider for constant value adjustment is not available in the tile view! Here it is possible to adjust values by the step width configured in the ETS by means of a short or long press of the buttons "Curtain position down" and "Curtain position up".

The function can be executed in a time-controlled manner using the "Timer" button. The curtain heights available range from 0 % to 100 % in 10 % increments.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The status value for the curtain position [F] can be shown in the Gira G1. The status value for the curtain position [F] is received by the "Feedback curtain position" communication object. The status value is visualised depending on the object value of the curtain position feedback from the shutter actuator being addressed. Position values are shown in plain text (0...100%). The status value is displayed as "---" for as long as no feedback object value has been received after a device reset.

A slider [E] is shown in the display element in the detail view of the "Shutter positioning" channel function. This slider shows position values in graphical format using the position value feedback from the shutter actuator being addressed. The position of the slider increases as the curtain is raised. Similarly, the slider [E] decreases to up to 100% as the curtain lowers. The slider [E] is always visible in the detail view and complements the "Status value" display.

The displayed function icon for this channel function can be configured in the ETS. The display icons shown at the top and bottom end of the slider [E] are fixed and cannot be changed. They show the user the reaction that is to be expected when the slider is moved in the corresponding direction.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Shutter positioning" function has five KNX communication objects:

- "Channel n - curtain short-term operation (stop)" (1-bit transmitting):
  When you press the "STOP" button in the detail view or briefly tap the "Curtain position" button in the tile view, this object stops a moving curtain in its current position.
- "Channel n - curtain long-term operation" (1-bit transmitting): When you touch the button for an extended time, this object emits long-term telegrams (up, down) to the KNX. This means that controlled curtains of shading systems can be moved to the end positions. KNX blind or shutter actuators execute the configured long-term operation (move). A short-term telegram (see "Channel n - curtain short-term operation (stop)" object (1-bit transmitting)) can interrupt ongoing drive configurations of long-term operation (stop), making it possible to halt a curtain at any desired point.
- "Channel n - feedback drive movement" (1-bit receiving): Through this object, the actuator is able to tell the device that the controlled drive (e.g. a shutter) is in motion.
  The position value must be fed back through the "active transmitting" actuator. This object is only available if the feedback function is configured.
- "Channel n - curtain position" (1-byte transmitting):
  When you press the buttons or operate the slider, this object emits value telegrams (0...100%) for curtain height specification to the KNX. This makes it possible to control a curtain position object (e.g. "Position blind", "Position shutter/awning", "Position ventilation flap"...) of a shutter or blind actuator.

- "Channel n - feedback curtain position" (1-byte receiving): This object allows the device to receive a position feedback message regarding curtain height, e.g. emitted by a shutter actuator. This influences the status value and slider for curtain height visualisation. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly.
  The position value must be fed back through the "active transmitting" actuator.

The communication objects interpret values to be transmitted or received based on KNX data point type 5.001 (Scaling). Decimal data values 0...255 are interpreted as percentage values 0...100%.

The value of the object "Channel n - feedback curtain position" is not evaluated during operation. Only once you are finished operating the control do the display elements react to the feedback object value from the shutter actuator again.

### 6.2.12. Room temperature

With the "Room temperature" channel function, the Gira G1 can be used as an extension of a room temperature controller. A controller extension is not involved in the temperature control itself. It allows the user to operate single-room control, i.e. a controller main unit, from various places in the room. This means that any number of operating extensions can be set up. External room temperature controllers (controller main units) that work with absolute value specifications can be fully controlled. The state of the controller main unit can be fully rendered in the Gira G1 (heating message, cooling message, setpoint specification, room temperature, setpoint temperature and current operating mode, heating and cooling mode toggle).

#### Connection to the room temperature controller

Typical KNX room temperature controllers generally offer various options for influencing or visualising room temperature control.

- Toggling between different operating modes (e.g. "Comfort", "Night"...) assigned different setpoint temperatures in the controller.
- Signalling whether there is anyone in the room. This may also be associated with a parameterised operating mode changeover in the controller.
- Adjusting the setpoint temperature by the parameterised step width (0.1 K, 0.2 K, 0.5 K or 1.0 K).

The parameterised step width on the controller extension must match the step width of the controller main unit.

The "Room temperature" channel function is operated in the device's user interface. The function is given a tile or detail view. It is possible to completely control a room temperature controller by changing the operating mode, by specifying the presence situation or by adjusting the setpoint value.

The "Room temperature" channel function of the Gira G1 is only intended for use on KNX compliant room temperature controllers. Functional limitations may arise when using other room temperature controllers.

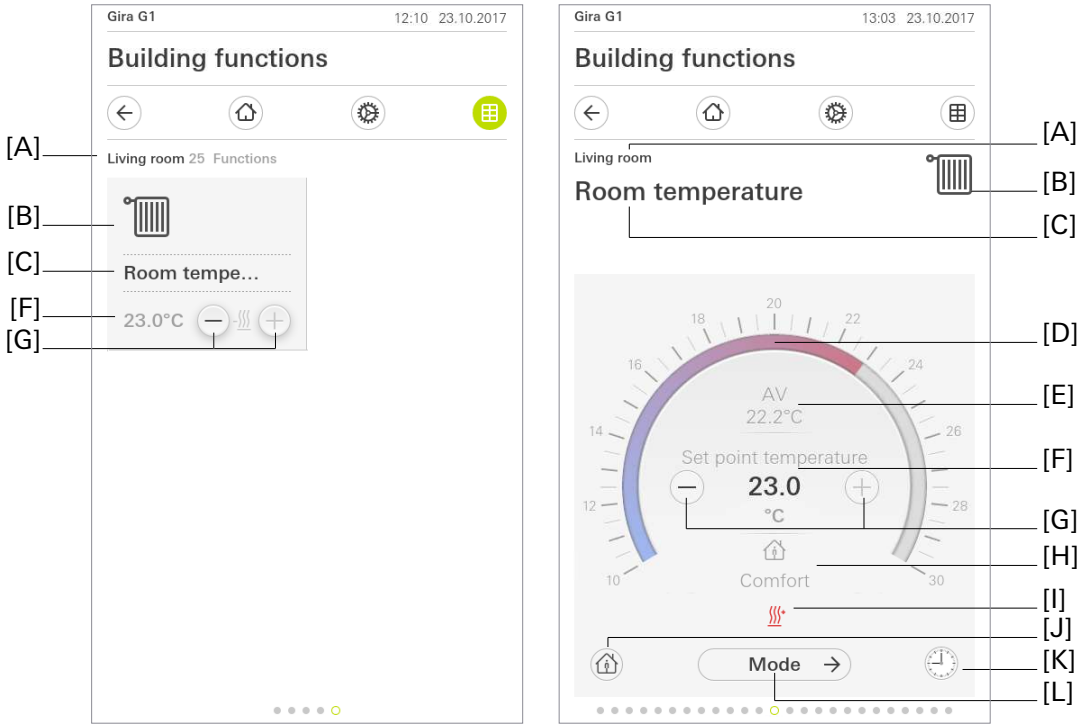


Fig 16: Example of operating and display element of the "Room temperature" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Operable scale
[E] Display of actual temperature
[F] Display of setpoint temperature
[G] "Value specification" button
[H] Display of active operating mode
[I] Display of heating/cooling
[J] "Presence" button
[K] "Timer" button
[L] "Mode" button

The "Room temperature" channel function has a display function alongside the operating function. Various pieces of status information concerning temperature control can be displayed in the user interface of the Gira G1. Because the statuses and information displayed and some operating functions depend heavily on how the controller main unit is parameterised, the controller extension must also be parameterised and matched to the functions of the controller main unit. This matching of functions takes place through parameters in the "Room temperature controller (RTC)" parameter node.
The function can be executed in a time-controlled manner using the "Timer" button. The operating modes available are "Comfort", "Standby", "Night" and "Frost/heat protection".

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

The "Room temperature" function has up to eleven KNX communication objects:

- "Channel 1 - actual temperature" (2-byte receiving):
  2-byte object for receiving the measured temperature value from the controller main unit via the KNX.
- "Channel 1 - setpoint temperature" (2-byte transmitting):
  2-byte object for externally specifying a setpoint value with absolute setpoint value specification. The possible value range is limited in dependence on the operating mode by the parameterised frost protection and/or heat protection temperature. The controller rounds the temperature values received through the object to the specified "step width of the setpoint offset". This depends on the application program used. The step width can be configured in the ETS (0.1 K / 0.2 K / 0.5 K / 1.0 K). The temperature value must always be specified in the "°C" format.
- "Channel 1 - feedback setpoint temperature" (2-byte receiving):
  2-byte object for receiving the current setpoint temperature value via the KNX.
- "Channel 1 - operating mode changeover" (1-byte transmitting):
  1-byte object for changeover of the controller main unit operating mode in accordance with the KNX specification.
- "Channel 1 - status of operating mode" (1-byte receiving):
  1-byte object with which the Gira G1 receives the current operating mode. This object is used to ensure that controller extensions are able to display the controller operating mode correctly in the KNX compliant status display. Consequently, this object must be connected with the controller main unit.

- "Channel 1 - presence object" (1-bit transmitting):
  This communication object is only enabled when the "Presence detection with presence button?" parameter is set to active. Presence is transmitted to the controller main unit via a button in the Gira G1 user interface. Absence can also be reported to the Gira G1 by the controller main unit. An active presence is displayed in the user interface.
  Polarity: presence = "1", no presence = "0".
- "Channel 1 - feedback presence object" (1-bit receiving):
  1-bit object for receiving presence. Polarity: presence = "1", no presence = "0"
- "Channel 1 - heating message" (1-bit receiving):
  1-bit object for receiving messages from the controller main unit as to whether heat energy is requested. Object value = "1": energy request, object value = "0": no energy request.
- "Channel 1 - cooling message" (1-bit receiving):
  1-bit object for receiving messages from the controller main unit as to whether cooling energy is requested. Object value = "1": energy request, object value = "0": no energy request.
- "Channel 1 - heating / cooling toggle" (1-bit transmitting):
  1-bit object for toggling the operating modes of heating/cooling in the controller main unit
- "Channel 1 - feedback heating/cooling" (1-bit receiving):
  1-bit object with which the Gira G1 receives the current operating mode (heating/cooling) of the controller main unit

The status elements of the controller extension are influenced by the communication objects.

### Operating mode changeover

The controller operating mode can be changed with a 1-byte communication object in accordance with the standard function block for room temperature controllers defined in the KNX manual. The object "Channel 1 - operating mode changeover" allows you to choose between the modes:

- Comfort mode
- Standby mode
- Night mode
- Frost/heat protection mode

The operating mode that is sent out to the KNX via the controller extension is defined in a selection list in the user interface of the Gira G1, which can be opened by tapping the "Mode" button. The desired operating mode is selected and sent to the KNX upon pressing the "OK" button. The operating mode fed back through the "KNX status operating mode" communication object is visualised in the detail view [G]. This object must be connected with the object in the controller main unit with the same function. If standby, night or frost/heat protection mode is activated, the controller extension of the Gira G1 shows eco mode.

### Operating type changeover

The operating type changeover can be implemented for room temperature controllers with a 1-bit communication object. The object "Channel 1 - heating/cooling changeover" allows you to choose between the operating types:

- Heating mode
- Cooling mode

The operating type that is sent out to the KNX via the "Room temperature" channel function is defined in a selection list in the user interface of the Gira G1, which can be opened by tapping the "Mode" button (Fig. 16). The desired operating type is selected and sent to the KNX upon pressing the "OK" button. The operating type fed back through the "Feedback heating/cooling" communication object is visualised in the detail view.

### Presence detection

If the room temperature controller works with a presence object for presence detection, presence can also be signalled by the "Room temperature" channel function of the Gira G1. The "Presence" button (Fig. 16) is only visible when "Presence detection with presence button" is activated (see "Presence detection with presence button?" parameter).

The "Presence object" communication object is enabled. The button [J] is internally connected to the "Presence object". The Gira G1 sends a "1" to the KNX when the button [J] is active. The "Presence" button displays in green in the case of active presence. Pressing the button [J] again sends a "0" to the KNX, therefore deactivating the presence.

The "Presence object" communication object makes it possible to receive the presence status from the room temperature controller and display it in the user interface of the Gira G1.

The reaction of the room temperature controller to telegrams from this communication object must be set in the parameters of the room temperature controller.

### Setpoint value specification (absolute setpoint adjustment)

Another function of the "Room temperature" channel function is the setpoint value specification. The "Value adjustment" buttons [G] (Fig. 16) make it possible to specify the setpoint value of the active operating mode on a room temperature controller.

Each time it is pressed, the button [G] reduces or increases the value of the setpoint by the "step width of the setpoint offset" set in the parameters of the Gira G1. The parameterised step width on the controller extension must match the step width of the controller main unit. The setpoint value emitted by the controller main unit is always displayed in the Gira G1 user interface.

The setpoint value specification of the controller extension of the Gira G1 can only be used if the controller main unit also works with absolute setpoint value specification.

### Communication with the controller main unit:

So that the controller extension can specify the setpoint value on a room temperature controller, the controller must have input and output objects for specifying the setpoint value of the active operating mode. The controller object that transmits the temperature setpoint value to the KNX must be connected with the "Setpoint temperature" input object of the controller extension and the controller object that receives temperature setpoint values must be connected with the "Setpoint value active operating mode" output object of the controller extension of the Gira G1, through an independent group address.

### Display of heating and cooling messages

The "Room temperature" channel function can show in the Gira G1 user interface whether heating or cooling energy is currently being requested. The display takes the form of the icons for heating or for cooling.

So that the display works, the message objects for heating and/or cooling of the "Room temperature" channel function of the Gira G1 and the controller must be connected.

### Behaviour after a device restart

The different display and operating functions of the "Room temperature" channel function are controlled by means of various communication objects. So that all status information is valid and present upon initialisation of the controller extension after a programming operation or upon return of the power supply, a controller main unit must communicate current statuses to the controller extensions, i.e. update the communication objects. For some objects, this takes place automatically during initialisation of the main unit.

First, all controller extensions should be commissioned as part of start-up. Only then should the controller main unit be connected and programmed.

### 6.2.13. Room temperature ON/OFF

The channel function "Room temperature ON/OFF" (controller extension switchable) has the same functional scope as the "Room temperature" channel function. It is also possible to activate or deactivate a controller main unit via an "ON/OFF" button (1-bit object)

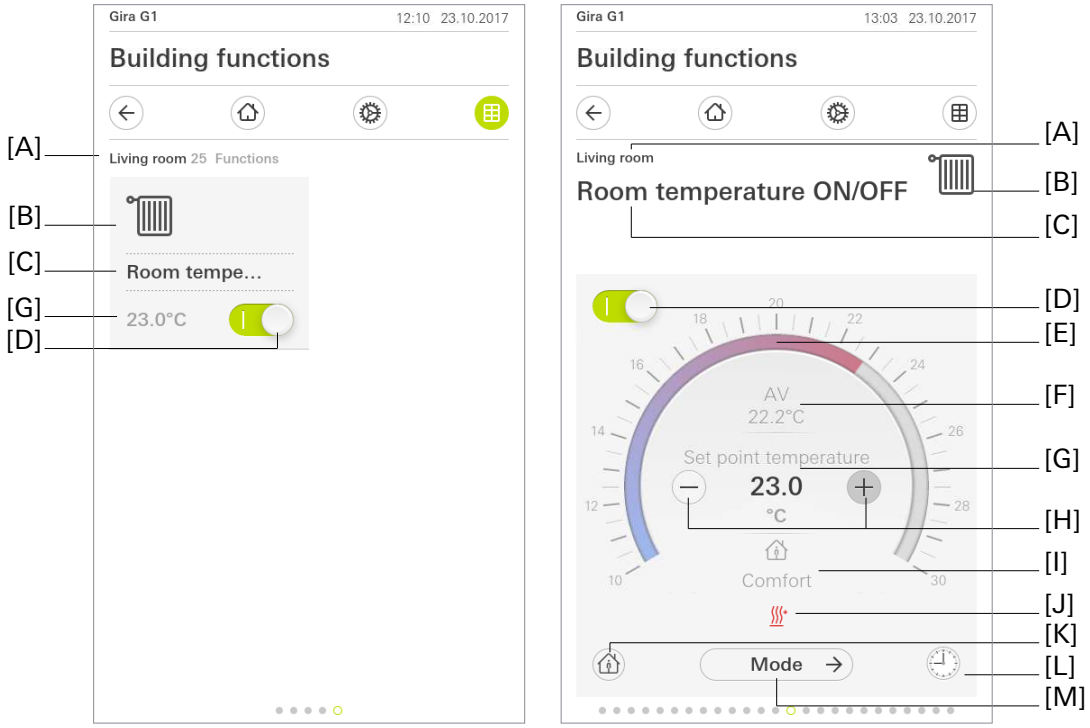


Fig 17: Example of operating and display element of the "Room temperature ON/OFF" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Button (ON / OFF)
[E] Operable scale
[F] Display of actual temperature
[G] Display of setpoint temperature
[H] "Value specification" button
[I] Display of active operating mode
[J] Display of heating/cooling
[K] "Presence" button
[L] "Timer" button
[M] "Mode" button

When using the "Room temperature ON/OFF" channel function (controller extension switchable), a slider switch is available as an additional operating and display element. This switch is used to activate or deactivate the controller main unit or display the status of the controller main unit (activated or deactivated). The function can be executed in a time-controlled manner using the "Timer" button. The operating modes available are "Comfort", "Standby", "Night" and "Frost/heat protection".

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

The "Room temperature" function has up to 13 KNX communication objects:

- "Channel 1 - actual temperature" (2-byte receiving):
  2-byte object for receiving the measured temperature value from the controller main unit via the KNX.
- "Channel 1 - setpoint temperature" (2-byte transmitting):
  2-byte object for externally specifying a setpoint value with absolute setpoint value specification. The possible value range is limited in dependence on the operating mode by the parameterised frost protection and/or heat protection temperature. The controller rounds the temperature values received through the object to the specified "step width of the setpoint offset". This depends on the application program used. The step width can be configured in the ETS (0.1 K / 0.2 K / 0.5 K / 1.0 K). The temperature value must always be specified in the "°C" format.
- "Channel 1 - feedback setpoint temperature" (2-byte receiving):
  2-byte object for receiving the current setpoint temperature value via the KNX.
- "Channel 1 - operating mode changeover" (1-byte transmitting):
  1-byte object for changeover of the controller main unit operating mode in accordance with the KNX specification.
- "Channel 1 - status of operating mode" (1-byte receiving):
  1-byte object with which the Gira G1 receives the current operating mode. This object is used to ensure that controller extensions are able to display the controller operating mode correctly in the KNX compliant status display. Consequently, this object must be connected with the controller main unit.
- "Channel 1 - presence object" (1-bit transmitting):
  This communication object is only enabled when the "Presence detection with presence button?" parameter is set to active. Presence is transmitted to the controller main unit via a button in the Gira G1 user interface. Absence can also be reported to the Gira G1 by the controller main unit. An active presence is displayed in the user interface. Polarity: presence = "1", no presence = "0".
- "Channel 1 - feedback presence object" (1-bit receiving):
  1-bit object for receiving presence. Polarity: presence = "1", no presence = "0"
- "Channel 1 - heating message" (1-bit receiving):
  1-bit object for receiving messages from the controller main unit as to whether heat energy is requested. Object value = "1": energy request, object value = "0": no energy request.
- "Channel 1 - cooling message" (1-bit receiving):
  1-bit object for receiving messages from the controller main unit as to whether cooling energy is requested. Object value = "1": energy request, object value = "0": no energy request.
- "Channel 1 - heating / cooling toggle" (1-bit transmitting):
  1-bit object for toggling the operating modes of heating/cooling in the controller main unit
- "Channel 1 - feedback heating/cooling" (1-bit receiving):
  1-bit object with which the Gira G1 receives the current operating mode (heating/cooling) of the controller main unit
- Switching (1-bit transmitting):
  1-bit object for activating or deactivating (or blocking) the controller main unit.
- Feedback switching (1-bit receiving):
  1-bit object with which the Gira G1 receives the activated or deactivated status of the controller main unit

### 6.2.14. Sauna temperature

The sauna temperature (sauna extension) has a display function alongside the operating function. The current temperature and setpoint temperature can be displayed in the user interface of the Gira G1.

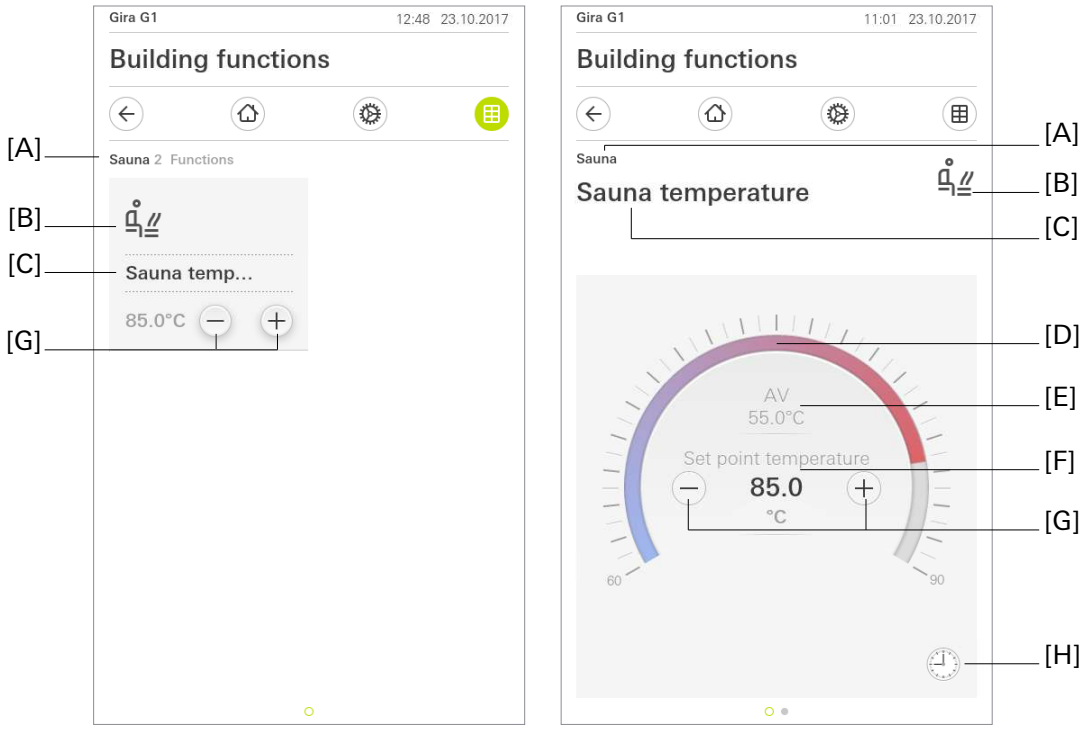


Fig 18: Example of operating and display element of the "Sauna temperature" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Operable scale
[E] Display of actual temperature
[F] Display of setpoint temperature
[G] "Value adjustment" button
[H] "Timer" button

Telegrams that change the setpoint temperature are sent to the KNX by pressing the "Value adjustment" button [G] in the tile view.

In the ETS, the value range for the setpoint temperature can be selected based on the sauna being controlled. The ambient temperatures available for the operable scale [D] are "40...70 °C", "50...80 °C", "60...90 °C", "70...100 °C" and "80...110 °C".

The step widths of "1K", "5K" and "10K" are available for value adjustment [G] in the ETS.

The function can be executed in a time-controlled manner using the "Timer" button. The options are the configured ambient temperatures (40...70°C), (50...80°C), (60...90°C), (70...100°C) or (80...110°C) in step widths of 5°C.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The actual temperature [E] and setpoint temperature [F] are visible in the detail view. At the same time, the temperature is shown on the operable scale [D].

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

The sauna temperature function has three KNX communication objects:

- "Channel n - actual temperature" (2-byte receiving):
  2-byte object for receiving the measured temperature value from the sauna via the KNX.
- "Channel n - setpoint temperature" (2-byte transmitting):
  2-byte object for transmitting the current setpoint temperature value via the KNX.
- "Channel n - feedback setpoint temperature" (2-byte receiving):
  2-byte object for receiving the current temperature setpoint value via the KNX. The temperature value is always given in °C.

### 6.2.15. Sauna temperature ON/OFF

The sauna temperature (sauna extension) has a display function alongside the operating function. The current temperature and setpoint temperature can be displayed in the user interface of the Gira G1.

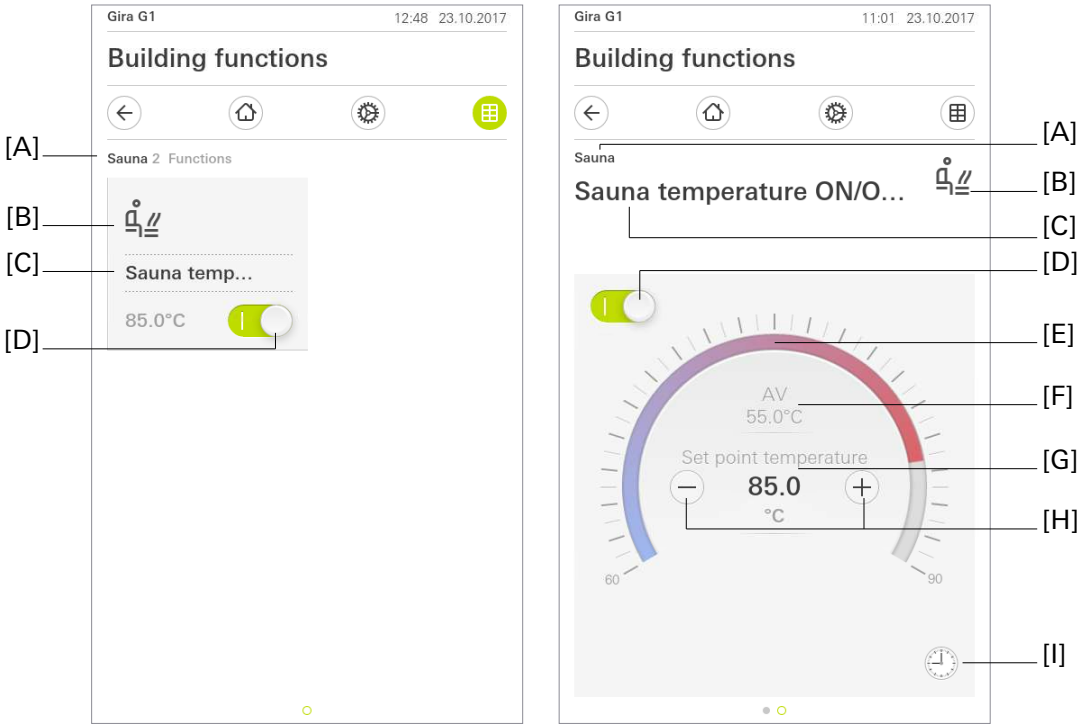


Fig 19: Example of operating and display element of the "Sauna temperature ON/OFF" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Button (ON / OFF)
[E] Operable scale
[F] Display of actual temperature
[G] Display of setpoint temperature
[H] "Value adjustment" buttons
[I] "Timer" button

The On/Off switch icon in both the tile and detail views [D] allows you to implement a switching process to switch the sauna being controlled on and off. You can only transmit switching telegrams to the KNX when operating in the tile view. Modifying the setpoint temperature requires you to switch to the detail19, view.

Telegrams that change the setpoint temperature are sent to the KNX by pressing the "Value adjustment" button [H] in the detail view.

In the ETS, the value range for the setpoint temperature can be selected based on the sauna being controlled. The ambient temperatures available for the operable scale [E] are "40...70 °C", "50...80 °C", "60...90 °C", "70...100 °C" and "80...110 °C".

The step widths of "1K", "5K" and "10K" are available for value adjustment [H] in the ETS.

The function can be executed in a time-controlled manner using the "Timer" button. The options are "switch on", "switch off" and the configured ambient temperatures (40...70°C), (50...80°C), (60...90°C), (70...100°C) or (80...110°C) in step widths of 5°C.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The actual temperature [F] and setpoint temperature [G] are visible in the detail view. At the same time, the temperature is shown on the operable scale [E].

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

The sauna temperature function has five KNX communication objects:

- "Channel n - actual temperature" (2-byte receiving):
  2-byte object for receiving the measured temperature value from the sauna via the KNX.
- "Channel n - setpoint temperature" (2-byte transmitting):
  2-byte object for transmitting the current setpoint temperature value via the KNX.
- "Channel n - feedback setpoint temperature" (2-byte receiving):
  2-byte object for receiving the current temperature setpoint value via the KNX. The temperature value is always given in °C.
- "Channel n - switching" (1-bit transmitting):
  1-bit object for transmitting the current switching condition via the KNX.
- "Channel n - feedback switching" (1-bit receiving):
  1-bit object for receiving the current switching condition via the KNX.

### 6.2.16. Air conditioning

The "Air conditioning" channel function makes it possible to control an air conditioning system through the Gira G1. You can toggle between the different operating modes. The desired temperature, level of ventilation and direction of air flow can be set.

It is important to note that the control only concerns the air conditioning system. The Gira G1 only takes over control of the air conditioning system.

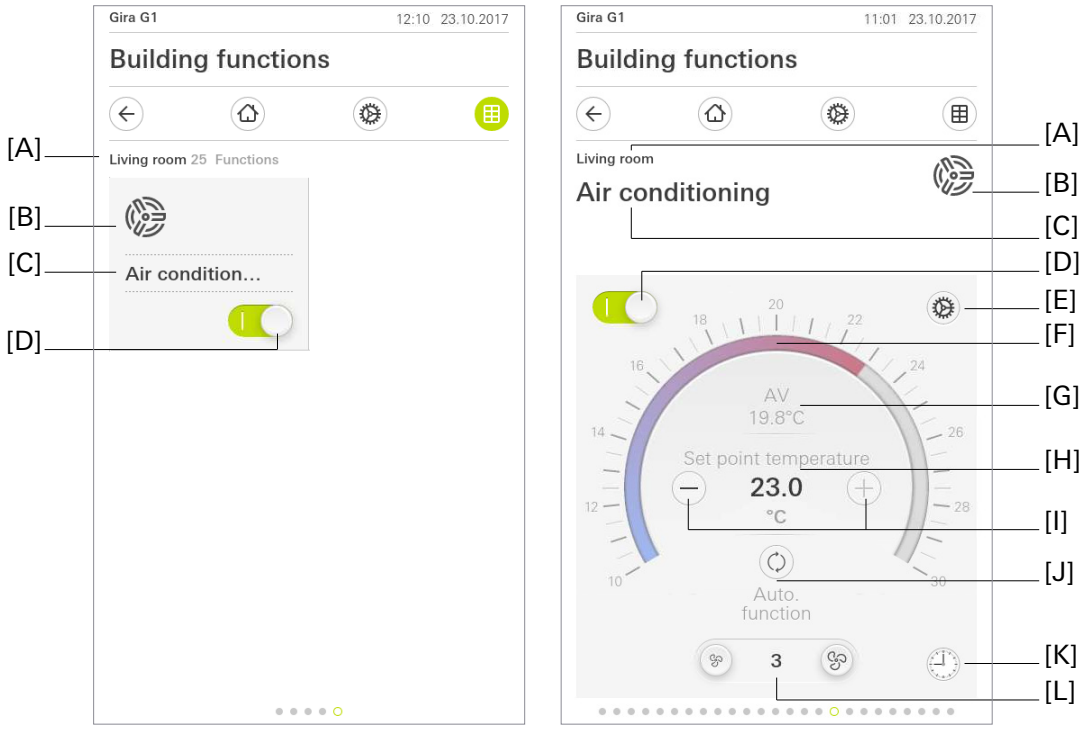


Fig 20: Example of operating and display element of the "Air conditioning" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Button (ON / OFF)
[E] "Settings" button
[F] Operable scale
[G] Display of actual temperature
[H] Display of setpoint temperature
[I] "Value specification" buttons
[J] "Mode" button
[K] "Timer" button
[L] Fan controller display and buttons

Air conditioning control can be activated using the On/Off switch icon in both the tile and detail view [D]. If this icon is deactivated, nothing can be changed in any other settings. Changes are only permitted once the On/Off switch icon has been activated.

Briefly pressing ⊕ or ⊖ [I] in the detail view allows you to increase or decrease the setpoint temperature by the parameterised step width. The new setpoint temperature is then sent directly to the bus. If you press the ⊕ or ⊖ button for an extended time, the setpoint temperature is moved up or down at a defined interval. The modified setpoint temperature is sent to the bus immediately, even if the button is still being pressed.

The setpoint temperature may also be specified by means of an operable scale [F]. Here, the setpoint temperature can be changed by dragging or tapping on the scale. With the "drag" procedure, the new setpoint temperature is only sent to the bus when you are finished operating the control.

The operating mode can be changed by pressing the button [J]. The following operating modes are available for selection here, depending on the parameterisation:

| | | |
|---|---|---|
|  | Automatic | The air conditioning system regulates independently according to the specified setpoint temperature. |
|  | Heating | The air conditioning system is only in heating mode |
|  | Cooling | The air conditioning system is only in cooling mode |
|  | Dehumidification | The air conditioning system ensures that the relative humidity in the room is reduced |
|  | Ventilation | The air conditioning system ventilates the room without heating/cooling it. |

The fan stages can be specified using the button [L]. There are up to 5 fan stages available, which can be changed gradually by means of "Fan down" or "Fan up". Alongside the 5 stages, the functions of "Max", "Auto" and "Stop" can be activated in the parameters. It is possible to gradually switch between these functions and the up to 5 stages by pressing "Fan down" or "Fan up". The fan stages can be displayed in percent as well as in individual stages. This can be changed in the parameters.

Changeover to percent affects the entire air conditioning channel. It is not possible to simply display fan stages in percent and air flow in stages, for example.

A warning may appear in the bottom left if an error message comes in. This is a sign that something is not right with the air conditioning system. Depending on the air conditioning system being used, it is possible to store an error text in this warning to be displayed upon touching the button by means of a 14-byte object.

Pressing the button [E] opens the settings in the air conditioning system. Under settings [E], there are various options for selection depending on the parameterisation:

- Change operating mode
- The air conditioning settings implemented can be stored using the "Save current settings" button. There are 5 free memory spaces available. Should they all be assigned, this is indicated when the "Save current settings" button is pressed. In this case, you must first delete entries under "Stored settings".
- The settings previously stored by the user can be accessed again under "Stored settings". It is also possible to delete entries.
- Depending on the design of the air conditioning system and parameterisation, the "Air flow horizontal" and/or "Air flow vertical" option may be used. It is possible to set the air flow in up to 5 individual steps or an oscillating movement. Alongside the settings of "Blade stages / Blade angle", you can set the air flow to oscillate by pressing the "Oscillate" button. During oscillation, the air flow is alternated between up/down, left/right. This can be stopped using the "Stop" button. How exactly the blade angles are moved depends on the air conditioning system, and can be specified therein.

The air flow setting is not available for every air conditioning system.

The function can be executed in a time-controlled manner using the "Timer" button. The options are "switch on" and "switch off", as well as all default values stored at the point of setting. When you trigger a default value, a "1" is first written to the switching object function. Directly after, the operating modes, fan stages and horizontal / vertical slat position corresponding to the default value are written.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

The setpoint temperature [H] transmitted by the air conditioning is shown in the detail view of the Gira G1. "---" is displayed if no telegram has yet been received after a restart. In addition, the setpoint temperature is displayed by means of a scale, which updates based on the feedback value for the setpoint temperature. The unit of setpoint temperature is displayed in °C.

The operating mode [J] updates depending on the feedback sent by the air conditioning system. In addition, various operating modes can be activated or deactivated in the parameters according to the availability of the air conditioning system.

The actual temperature is displayed centrally in the detail view [G]. This can be assigned to the corresponding channel through the "Actual temperature" communication object. "—" is displayed if no telegram has been received after a restart.

The fan stage is shown in the detail view [L]. The precise presentation is based on the parameterisation. The following presentations are possible, depending on the parameterisation: Percentages, individual steps, auto, max, stop. The size of the step width is specified in the parameters. Updates are based on feedback from the air conditioning system.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

- "Channel n – actual temperature" (2-byte receiving): This object makes it possible to display the current temperature in the room. It is shown in the detail view. Either the internal sensor of the G1 or an external source, such as the air conditioning system, can be used.
- "Channel n – setpoint temperature" (2-byte transmitting):
  This object is used to set the setpoint temperature in the air conditioning system. In the detail view, the setting can be implemented using the operating surfaces or scale. The setpoint temperature is modified in defined step widths. These can be changed in the parameters. The setpoint temperature can be varied within an upper and lower value specified in the parameters.
- "Channel n – feedback setpoint temperature" (2-byte receiving): This object updates the setpoint temperature in the tile view and the detail view. The scale is also depicted in the detail view through this object. The current setpoint temperature is transmitted to the G1 by the air conditioning system, depending on how it is parameterised.
- "Channel n – switching" (1-bit transmitting): This object switches the air conditioning system on or off. Polarity: "1" switched on, "0" switched off.
- "Channel n – feedback switching" (1-bit receiving): This object enables operation of the air conditioning function in the Gira G1. This is intended for remote release, e.g. directly from the air conditioning system. Polarity: "1" switched on, "0" switched off.
- "Channel n – operating mode changeover“ (1-byte transmitting):
  This object changes the operating mode in the air conditioning system. The following operating modes are available for selection:
  - Auto – value "0"
  - Heating – value "1"
  - Cooling – value "3"
  - Ventilation – value "9"
  - Drying – value "14"

  These are changed in the detail view or under settings. They may be switched off, depending on the parameterisation. (Not all air conditioning systems support all operating modes).

- "Channel n – status of operating mode" (1-byte receiving): This object updates the icons for the operating mode in the detail view. The icons hold the following values:
  - Auto – value "0"
  - Heating – value "1"
  - Cooling – value "3"
  - Ventilation – value "9"
  - Drying – value "14"

  These values are fed back by the air conditioning system.
- "Channel n – fan stage" (1-byte transmitting): This object transmits the currently selected fan stage to the air conditioning system in the form of absolute values. It is possible to select up to 5 fan stages. In the parameters, the value of the selected stage can be defined based on the air conditioning system. Max, auto and/or stop may also be defined as a fan stage.
- "Channel n – feedback fan stage" (1-byte receiving): This object updates the fan stage in the detail view. It is fed back by the air conditioning system in absolute values.
- "Channel n – fan speed" (1-byte transmitting):
  This object transmits the currently selected fan stage to the air conditioning system in the form of percentage values. It is possible to select up to 5 fan stages. In the parameters, the percentage to be sent for each stage can be defined based on the air conditioning system. Max, auto and/or stop may also be defined as a fan stage.
- "Channel n – feedback fan speed" (1-byte receiving): This object updates the fan stage in the detail view. It is fed back by the air conditioning system in percentage values.
- "Channel n – horizontal stage" (1-byte transmitting):
  This object influences the horizontal air flow of the air conditioning system in absolute values. This is possible in a maximum of 5 stages. The necessary values are specified in the parameters and may vary depending on the air conditioning system used. In addition, a value for constant alteration of the air flow and for stopping the movement can be defined in the parameters.
- "Channel n – feedback horizontal stage" (1-byte receiving): This object feeds back the absolute horizontal stage of the air conditioning's air flow. It is then updated in the settings.
- "Channel n – horizontal position" (1-byte transmitting): This object influences the horizontal air flow of the air conditioning system in percentage values. This is possible in a maximum of 5 stages. The necessary percentage values are specified in the parameters and may vary depending on the air conditioning system used. In addition, a percentage value for constant alteration of the air flow and for stopping the movement can be defined in the parameters
- "Channel n – feedback horizontal position" (1-byte receiving): This object feeds back the percentage horizontal position of the air conditioning's air flow. It is then updated in the settings.

The air flow setting is not available for every air conditioning system.

- "Channel n – horizontal stop/move" (1-bit transmitting): This object changes the air flow of the oscillating air conditioning system in horizontal direction. The polarity is: "1" move, "0" stop. How the "Move" command is interpreted is set in the air conditioning parameters. The object is only available if the parameter point "Show stages horizontal adjustment?" is inactive.
- "Channel n – feedback horizontal stop/move" (1-bit receiving): This object allows the air conditioning system to report whether the air flow is oscillating in horizontal direction. The polarity is: "1" move, "0" stop. The object is only available if the parameter point "Show stages horizontal adjustment?" is inactive.
- "Channel n – vertical stage" (1-byte transmitting): This object influences the vertical air flow of the air conditioning system in absolute values. This is possible in a maximum of 5 stages. The necessary values are specified in the parameters and may vary depending on the air conditioning system used. In addition, a value for constant alteration of the air flow and for stopping the movement can be defined in the parameters

- "Channel n – feedback vertical stage" (1-byte receiving): This object feeds back the absolute vertical stage of the air conditioning's air flow. It is then updated in the settings.
- "Channel n – vertical position" (1-byte transmitting): This object influences the vertical air flow of the air conditioning system in percentage values. This is possible in a maximum of 5 stages. The necessary percentage values are specified in the parameters and may vary depending on the air conditioning system used. In addition, a percentage value for constant alteration of the air flow and for stopping the movement can be defined in the parameters
- "Channel n – feedback vertical position" (1-byte receiving):
  This object feeds back the percentage vertical position of the air conditioning's air flow. It is then updated in the settings.

The air flow setting is not available for every air conditioning system.

- "Channel n – vertical stop/move" (1-bit transmitting): This object changes the air flow of the oscillating air conditioning system in vertical direction. The polarity is: "1" move, "0" stop. How the "Move" command is interpreted is set in the air conditioning parameters. The object is only available if the parameter point "Show stages vertical adjustment?" is inactive.
- "Channel n – feedback vertical stop/move" (1-bit receiving): This object allows the air conditioning system to report whether the air flow is oscillating in vertical direction. The polarity is: "1" move, "0" stop. The object is only available if the parameter point "Show stages vertical adjustment?" is inactive.
- "Channel n – error" (1-bit receiving):
  With this object, a warning symbol can be displayed in the detail view in the event that an error is reported by the air conditioning system. This serves as an indicator in order to allow problems to be identified as quickly as possible.
- "Channel n – error text" (14-byte receiving): This object communicates an error text from the air conditioning system to the G1. This error text can be brought up by pressing the warning symbol, which is controlled by the "Error" object. This provides a detailed error message from the air conditioning system, in order that action may be taken as quickly as possible.

### 6.2.17. Scene extension

Using the "Scene extension" channel function makes it possible to send out 1-byte telegrams as per KNX data point type 18.001 (SceneControl) via the KNX.

When combined with scene touch sensors or actuators that have a scene function, this makes it possible to control various KNX subsections and therefore set lighting and shading systems, for example, based on the situation. The channel function is always designed as a button function.

The "Functionality" parameter specifies how the scene extension works. This parameter has the following settings options:

- "Scene extension without save function"
- "Scene extension with save function"

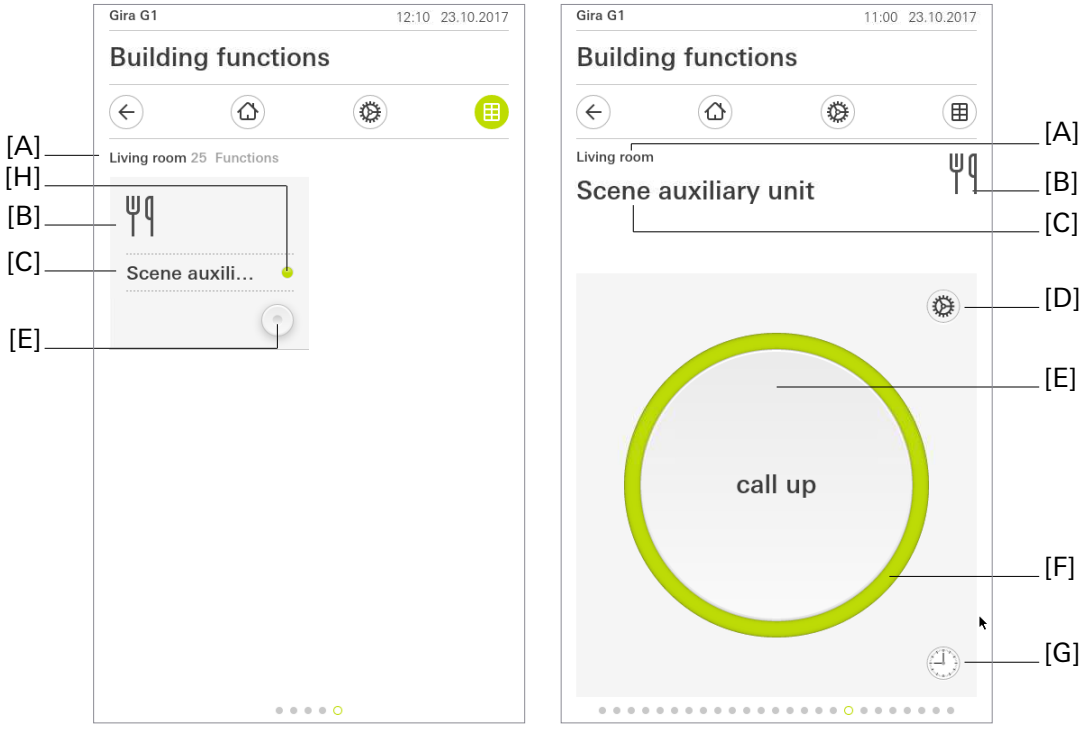


Fig 21: Example of operating and display element of the "Scene extension" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] "Settings" button
[E] Button to call up a scene number
[F] Status ring
[G] "Timer" button
[H] Status LED

Telegrams that transmit a preset scene number (1...64) to the KNX are generated by means of a button in the tile view. Tapping on the button to bring up a scene transmits the number that was configured as a scene number in the ETS to the KNX accordingly. This number is also sent upon any further command.

If the "Scene extension with save function" parameter is activated, the "Settings" button is enabled in the detail view. This button is used to open the settings for the channel function, which can be closed again using the "OK" button after saving. In the settings, the "Save scene" button transmits a save telegram to the KNX.

The scene can be called up in a time-controlled manner using the "Timer" button.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

### Status elements

A status text or value cannot be configured for the "Scene extension" channel function. Confirmation of transmission is provided in the form of the status elements [F] and [G] flashing green.

### Communication objects

The name of the communication object is based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Scene extension" function has one KNX communication object:

- "Channel n - scene extension" (1-byte transmitting): When you touch the button, this object emits scene extension telegrams (values as per KNX DPT 18.001) to the KNX. This makes it possible to directly control a scene touch sensor, a scene controller or an actuator with its own scene function, for example, through the extension object available on these devices. It is possible to emit call or save telegrams.

Status objects do not exist as a KNX channel with the "Scene extension" function does not have any dynamic status elements.

### 6.2.18. Value transmitter

The following value transmitter types can be used with the value transmitter channel function:

**Scroll-based value input:**

- 0...100% (1 byte / KNX 5.001)
- 0...255% (1 byte / KNX 5.004)
- 0...255 (1 byte / KNX 5.010)
- -128...127 (1 byte / KNX 6.010)
- Temperature (2 byte / KNX 9.001)

**Keypad-based value input**

- 0...255 with keypad input (1 byte / KNX 5.010)
- 0...65535 with keypad input (2 byte / KNX 7.001)
- 0...4294967295 with keypad input (4 byte / KNX 12.001)
- -128...127 with keypad input (1 byte / KNX 6.010)
- -32768...32767 with keypad input (2 byte / KNX 8.001)
- -2147483648..2147483647 with keypad input (4 byte / KNX 13.001)
- Decimal with keypad input (2 byte / KNX 9.x)
- Decimal with keypad input (4 byte / KNX 14.x)

The value transmitter channel function makes it possible to generate telegrams in accordance with KNX data point types 5.001, 5.004, 5.010, 6.010, 7,001, 8.001, 9.x, 12.001, 13.001 and 14.x, depending on the type of value transmitter. Value inputs on the Gira G1 may be restricted by means of limits (minimum and maximum value) configured through the ETS.

The basic value can be defined. This means that, for example, by controlling other KNX devices, the user can execute limit specifications or counter value specifications or specify temperature values.

Thanks to the identical data format, it is also possible to control dimming actuators (through brightness value specification) or blind and shutter actuators (through position value specification) as an alternative or in addition to the "Dimming (brightness value)" or "Blind/shutter (position)" channel functions.

For example, static brightness or position values can be configured and called up using buttons. Such control is an option if value adjustment by means of a long button press or visualisation of slat positions on a slider is not necessary.

The type of function icon shows the user directly on the device which function will be executed with the value transmitter. The step width can be configured through the ETS for the scroll-based value transmitter. In terms of decimal value transmitters with keypad-based input it is possible to configure how many decimal places are permitted.

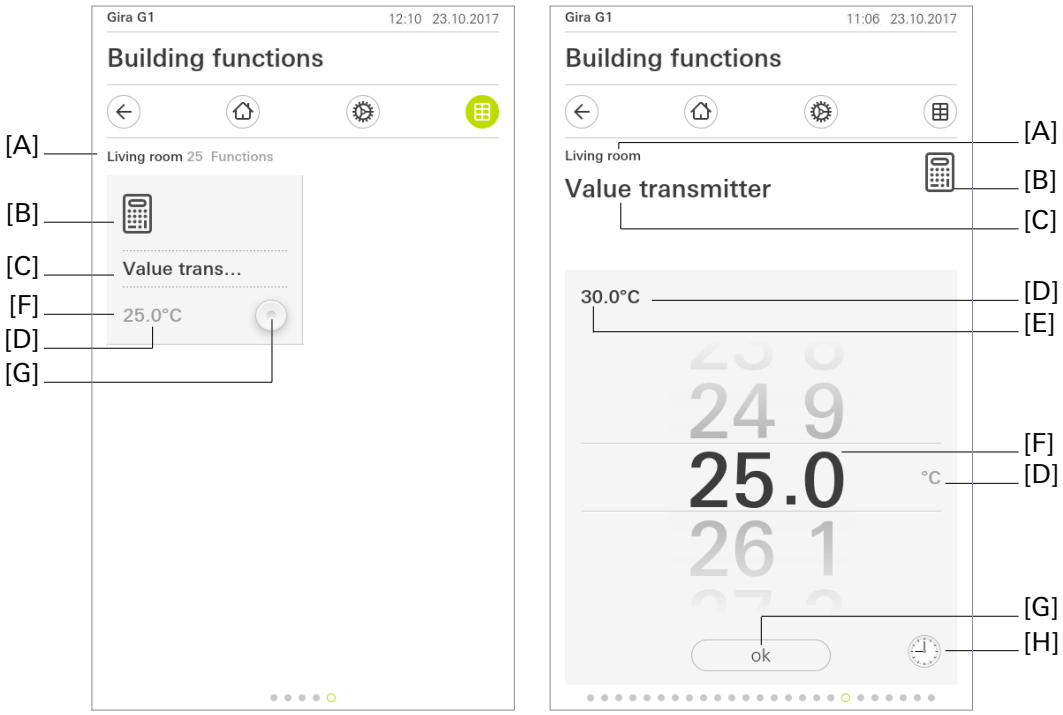


Fig 22: Example of operating and display element of the "Value transmitter (scroll)" function

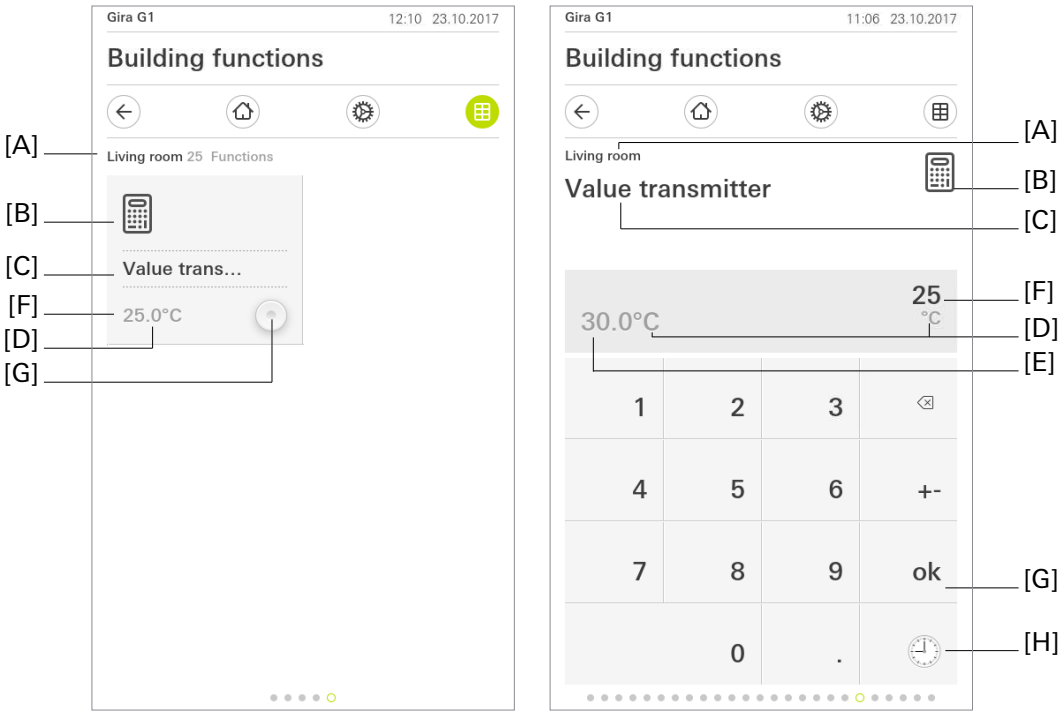


Fig 23: Example of operating and display element of the "Value transmitter (keypad)" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of "Unit"
[E] Display of "Status value", "Basic value"
[F] Selection / display of "Value input"
[G] "Value transmitter" / "OK" button
[H] Timer button

Telegrams that transmit the corresponding value to the KNX are generated by means of a button or entered using the keypad in the tile view. Tapping on the "Value transmitter" button [G] transmits the value that was configured or last set as the basic value to the KNX. This value is also transmitted upon any further actuation.

If value adjustment is enabled in the parameters of the ETS, the basic value set in the ETS can be adjusted by inputting values in the detail view and then transmitting them to the KNX using the "OK" button [G]. Only values that lie within the permitted values (minimum value and maximum value) can be emitted.

This value, set in the detail view, is then also transmitted to the KNX upon pressing the button in the tile view. The modified value is emitted upon every actuation of the value transmitter until a new value is input or the Gira G1 is restarted. The basic value configured in the ETS is active again after a device restart.

The channel function is set to the value transmitter configured in the ETS and cannot be changed.

Values can be transmitted in a time-controlled manner using the "Timer" button. Different value ranges are available depending on the parameterised type of value transmitter.

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be saved.

### Status elements

The status value [E] can be shown in the detail view. The status value is received by the corresponding channel feedback object. The status value is visualised depending on the object value of the value transmitter feedback from the device being addressed. Returned values are shown in plain text (e.g. 22 °C). The status value is displayed as "---" in the detail view for as long as no feedback object value has been received after a device reset.

You can choose to not use the status value and unit display. The parameter "Allow value adjustment?" makes it possible to remove the option of value adjustment on site on the device.

If the "Allow value adjustment?" parameter is deactivated, the value of the value transmitter function to be transmitted may only be adjusted in the Gira G1 parameters in the ETS.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Value transmitter" function has up to two KNX communication objects:

- "Channel n - value transmitter (0...100%; 0...255%; 0...255; -128...127...)" (1, 2 or 4-byte transmitting): When you press the button, this object emits value telegrams to the KNX. This makes it possible to control a limit value object, a brightness value object or a curtain position object, for example.
- "Channel n - value transmitter feedback (0...100%; 0...255%; 0...255... )" (1, 2 or 4-byte receiving): This object allows the device to receive a value feedback message, e.g. emitted by an actuator being addressed. This influences the status value for value visualisation. Consequently, the feedback object must always be configured in an orderly manner so that the status displays of the channel element work correctly.
  The value must be fed back through the "active transmitting" actuator.

### 6.2.19. Status display

The "Status display" channel function is simply a display function. The Gira G1 receives value telegrams through the KNX, allowing the device to display temperature values, brightness values or text.

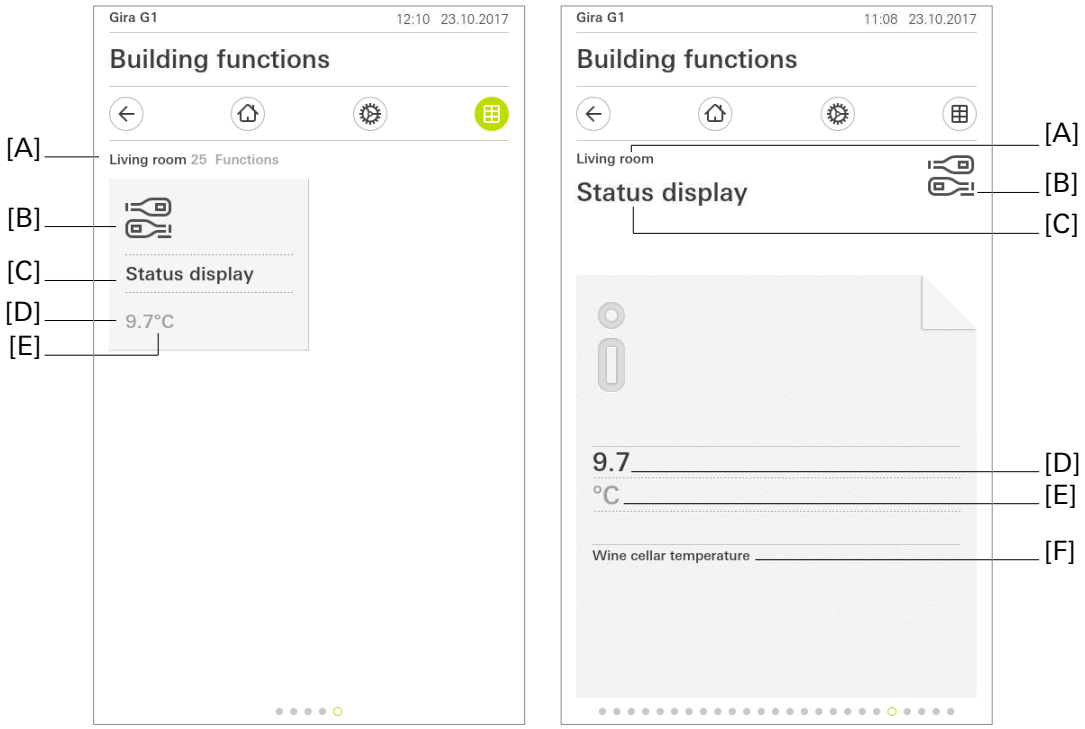


Fig 24: Example of operating and display element of the "Status display" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of "Status value"
[E] Display of "Unit"
[F] Display of "Additional text"

The "Status display" channel function makes it possible to receive 1-bit, 1-byte, 2-byte, 4-byte or 14-byte telegrams through the KNX and display them on the Gira G1. The telegrams correspond to different KNX data point types, depending on the display function:

| Display functions | Data point type |
|---|---|
| Status display | 1 bit |
| Value display 0...100% | 1 byte / KNX 5.001 |
| Value display 0...360° | 1 byte / KNX 5.003 |
| Value display 0... 255% | 1 byte / KNX 5.004 |
| Value display 0... 255 | 1 byte / KNX 5.010 |
| Value display -128...127 | 1 byte / KNX 6.010 |
| Value display 0...65535 | 2 byte / KNX 7.001 |
| Value display -32768...32767 | 2 byte / KNX 8.001 |
| Temperature display | 2 byte / KNX 9.001 |
| Brightness value display | 2 byte / KNX 9.004 |
| Value display 0...4294967295 | 4 byte / KNX 12.001 |

| Display functions | Data point type |
|---|---|
| Value display -2147483648...2147483647 | 4 byte / KNX 13.001 |
| Decimal value display | 4 byte / KNX 14.0xx |
| Text display | 14 byte / KNX 16.000 |

### Status elements

The received status value [D] is shown in both views. The values received from the KNX are shown in plain text (e.g. "22 °C").

In the case of a configured 1-bit status display, the presentation of the received object value ("1" or "0") in the Gira G1 user interface is based on the "Select status text" parameters for the switched-on and switched-off state. Text entered here is displayed in the user interface. A status LED can also be shown with this display function.

The status value is displayed as "---" for as long as no object value has been received after a device restart. The data format of this status value display is determined by the ETS parameter "Display function".

Depending on the setting of the "Display function" parameter, various parameters can be configured for the status elements:

- Colour of the status displays for O - OFF
- Colour of the status displays for I - ON
- Select status text for I - ON
- Select status text for O - OFF
- Display status LED
- Text for unit
- Number of decimal places
- Additional text

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Status display" function has one KNX communication object. The corresponding object is enabled depending on the set display function:

- "Channel n - status display" (1-bit receiving):
  This object allows the device to receive a 1-bit value feedback message sent out to the KNX. This influences the displayed value and the status LED, if these status elements are activated.
  The value must be actively transmitted to the KNX.
- "Channel n - value display (0...100%)“ (1-byte receiving):
  This object allows the device to receive a 1-byte value feedback message sent out to the KNX. The telegrams must be of a 1-byte data type. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - value display (0...360°)“ (1-byte receiving):
  This object allows the device to receive a 1-byte value feedback message sent out to the KNX. The telegrams must be of a 1-byte data type. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - value display (0...255%)“ (1-byte receiving):
  This object allows the device to receive a 1-byte value feedback message sent out to the KNX. The telegrams must be of a 1-byte data type. This influences the displayed value. The value must be actively transmitted to the KNX.

- "Channel n - value display (0...255)" (1-byte receiving):
  This object allows the device to receive a 1-byte value feedback message sent out to the KNX. The telegrams must be of a 1-byte data type. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - value display (-128...127)" (1-byte receiving):
  This object allows the device to receive a 1-byte value feedback message sent out to the KNX. The telegrams must be of a 1-byte data type. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - value display (0...65535)" (2-byte receiving):
  This object allows the device to receive a 2-byte value feedback message sent out to the KNX. The telegrams must be of a 2-byte data type. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - value display (-32768...32767)" (2-byte receiving): This object allows the device to receive a 2-byte value feedback message sent out to the KNX. The telegrams must be of a 2-byte data type. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - temperature display" (2-byte receiving):
  This object allows the device to receive a 2-byte value feedback message sent out to the KNX. The telegrams must be of data point type 9.001. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - brightness value display" (2-byte receiving):
  This object allows the device to receive a 2-byte value feedback message sent out to the KNX. The telegrams must be of data point type 9.004. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - value display (0...4294967295)" (4-byte receiving): This object allows the device to receive a 4-byte value feedback message sent out to the KNX. The telegrams must be of data point type 12.001. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - value display (-2147483648...2147483647)" (4-byte receiving):
  This object allows the device to receive a 4-byte value feedback message sent out to the KNX. The telegrams must be of data type 13.001. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - decimal value display" (4-byte receiving):
  This object allows the device to receive a 4-byte value feedback message sent out to the KNX. The telegrams must be of KNX data point type 14.0xx. This influences the displayed value. The value must be actively transmitted to the KNX.
- "Channel n - text display" (14-byte receiving):
  This object allows the device to receive a 14-byte text sent out to the KNX. The telegrams must be of KNX data point type 16.000. This procedure leads to the received text being displayed in the Gira G1 user interface. The text must be actively transmitted to the KNX.

### 6.2.20. IP camera

Using the "IP camera" channel function allows the video image of a network-enabled camera to be displayed in the user interface of the Gira G1.

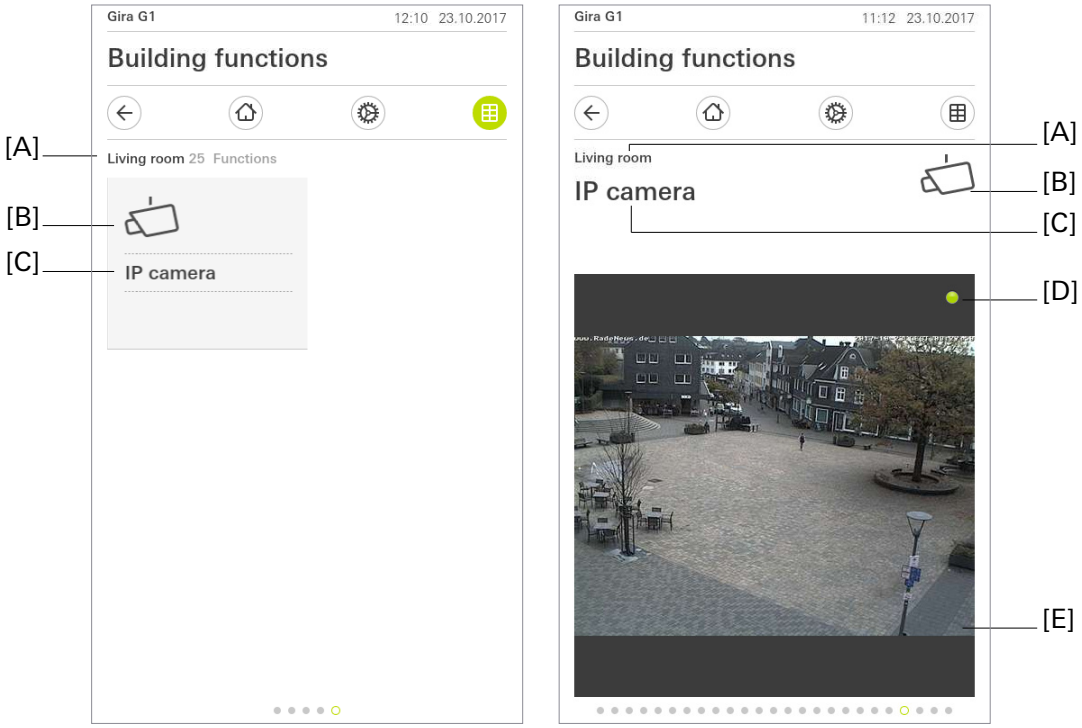


Fig 25: Example of operating and display element of the "IP camera" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Status LED
[E] Display of camera image

When you call up the "IP camera" channel function, a connection is established to the configured IP camera (status LED [D] is orange) and the video image is transmitted live once a connection has been established (status LED [D] is green).

Whilst the camera image is being watched, a KNX communication object can be used to implement switching actions, e.g. to inform occupants that the camera is being watched or to illuminate the area under surveillance. In addition, the switching telegram can be used elsewhere to execute any control task complying with the 1-bit data format in the KNX system. The "ON" telegram is transmitted upon attempting to connect up the camera image.

### Communication objects

A channel with the "IP camera" function has one communication object:

The name of the communication object is based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

- "Channel n - camera" (1-bit transmitting): When the "IP camera" channel function is opened in the detail view of the Gira G1 user interface, this communication object transmits a 1-bit switching telegram with the value "ON". If the user switches to another channel function in the user interface, moves to another view or switches off the screen of the Gira G1, a switching telegram with the value "OFF" is sent.

### Compatible transmission formats

- The camera image must be transmitted by the IP camera in M-JPEG format.
- HTTP and HTTPS with "Basic Authentication" is supported.
- IPv6 is not supported

It is advisable to limit the camera image for the Gira G1 in terms of resolution and frame rate in order to ensure smooth operation of the Gira G1. The recommended resolution is approx. 480x360, with a frame rate of 15 images per second.

### Examples for video URLs

MOBOTIX with password query via HTTPS:
https://user:password@192.168.0.111/cgi-bin/faststream.jpg
(Replace "user", "password" and "192.168.0.111" with your content)

AXIS with password query via HTTPS:
https://user:password@192.168.0.111/axis-cgi/mjpg/video.cgi?resolution=480x360
(Replace "user", "password" and "192.168.0.111" with your content)

Hikvision with password query via HTTP:
http://user:password@192.168.0.111/Streaming/channels/102/httppreview
(Replace "user", "password" and "192.168.0.111" with your content)
Note: In some models of Hikvision, M-JPEG is only possible in the configuration of a "substream", which is why a substream is referenced in the example URL.

The IP camera must be able to be reached through the Gira G1 network.

It is always recommended that you protect cameras from third-party access using a user name and password.

A HTTPS connection does not involve validation of the issuer of the certificate or the host name.

### 6.2.21. Open URL

You can open websites via a simple browser on the Gira G1 using the "Open URL" function. You can use this function to implement fast access to websites on the Gira G1.

Because the functions of the browser are restricted, please check the display and functionality of the set up website during start-up.

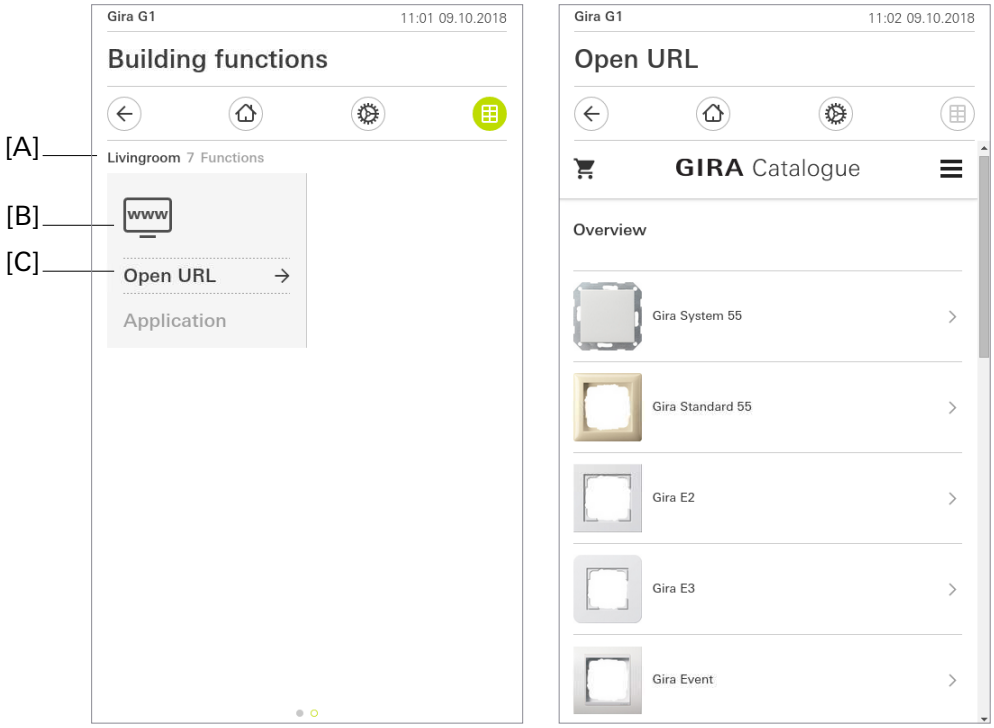


Fig 26: Example of operating and display element of the "Open URL" function

[A] Displays the "User interface description" function folder
[B] Displays the "Function icon"
[C] Displays the "User interface description" channel function

When you call up the "Open URL" channel function, a connection is established to the configured website and displayed once the connection has been established.

While the website is being viewed, switching actions can be executed using a KNX communication object.

#### Communication object

A channel with the "Open URL" function has one communication object:

The name of the communication object is based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

- "Channel n - Open URL active" (1 bit transmitting): When the "Open URL" channel function is opened in the detail view of the Gira G1 user interface, this communication object transmits a 1-bit switching telegram with the value "ON". If the user switches to another channel function in the user interface, moves to another view or switches off the screen of the Gira G1, a switching telegram with the value "OFF" is sent.

**Notes:**

- Please enter the website path, including protocol prefix, e.g., "http://...".
- Please check the display and functionality of the set up website during start-up because not all websites can be opened via the "Open URL" function.

**Examples**

The "Open URL" function, for example, can be used for the following scenarios:

- Websites created internally
  To display current information, such as the daily menu in restaurants or hotels.
  To create websites internally, please observe the information listed below.
- Websites of different devices
  To display pages that are generated on web servers of high-quality household devices, e.g., AV receivers, printers, "white goods" (washing machine, laundry dryer).
  Please note that device websites which do not fulfil the criteria listed below can only be displayed with restrictions or not at all.
- Digital picture frame
  To display images or other content, e.g., of local NAS systems.

**Information for internally created websites**

In order for a website to be optimally displayed via the "Open URL" function, the following criteria need to be fulfilled:

- It must be possible to display the application in an iframe.
- 480 x 681 pixels are displayed.
  If the area to be displayed is greater, scrollbars must be permitted by the page.
- No browser tabs or windows may be opened or controlled.
- No local links (anchors) may be used.
- JavaScript in links can lead to problems.
- Plug-ins (flash, video, etc.) are not displayed.

### 6.2.22. Audio Control

Using the audio function allows you to control hi-fi systems via KNX. The following functions are available: Play/pause track, change volume, mute and change between tracks (previous and next track).

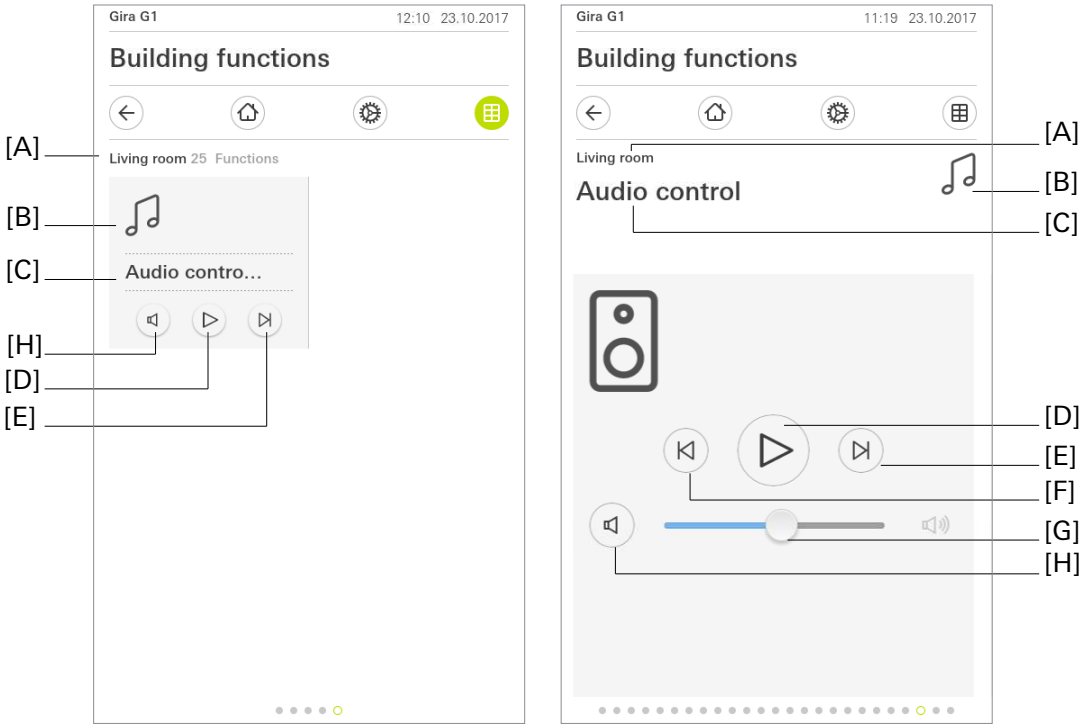


Fig 27: Example of operating and display element of the "Audio Control" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Play/Pause
[E] Next track
[F] Previous track
[G] Volume control
[H] Mute button

The Play/Pause button [D] allows you to start or pause playback of a track. The buttons [E] and [F] allow you to change track.

In addition, the volume can be controlled using a slider [G] or switched off completely using the "Mute" button [H].

The Play/Pause button does not show the current status, but the next possible status. You see the pause icon whilst the music is playing, and vice versa.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Audio Control" function has eight communication objects:

- "Channel n - playback (1-bit):
  1-bit object for starting playback. Polarity: playback active = "1", playback not active = "0"
- "Channel n - feedback playback" (1-bit):
  1-bit feedback object for transmitting the current playback status. Polarity: playback active = "1", playback not active = "0"
- "Channel n - volume" (1-byte):
  1-byte object for setting the volume. Volume between 0% (quiet) and 100% (loud).
- "Channel n - feedback volume" (1-byte):
  1-byte feedback object for reading the current volume. Volume between 0% (quiet) and 100% (loud).
- "Channel n - mute" (1-bit):
  1-bit object for muting. Polarity: muted = "1", not muted = "0".
- "Channel n - feedback muting" (1-bit):
  1-bit feedback object for reading the current muting status.
- "Channel n - previous track" (1-bit):
  1-bit object for playing the previous track.
- "Channel n - next track" (1-bit):
  1-bit object for playing the next track.

### 6.2.23. Audio control with media data

Using the audio function allows you to control hi-fi systems via KNX. The following functions are available: Play/pause track, change volume, mute, change between tracks (previous and next track) and display track, artist and album.

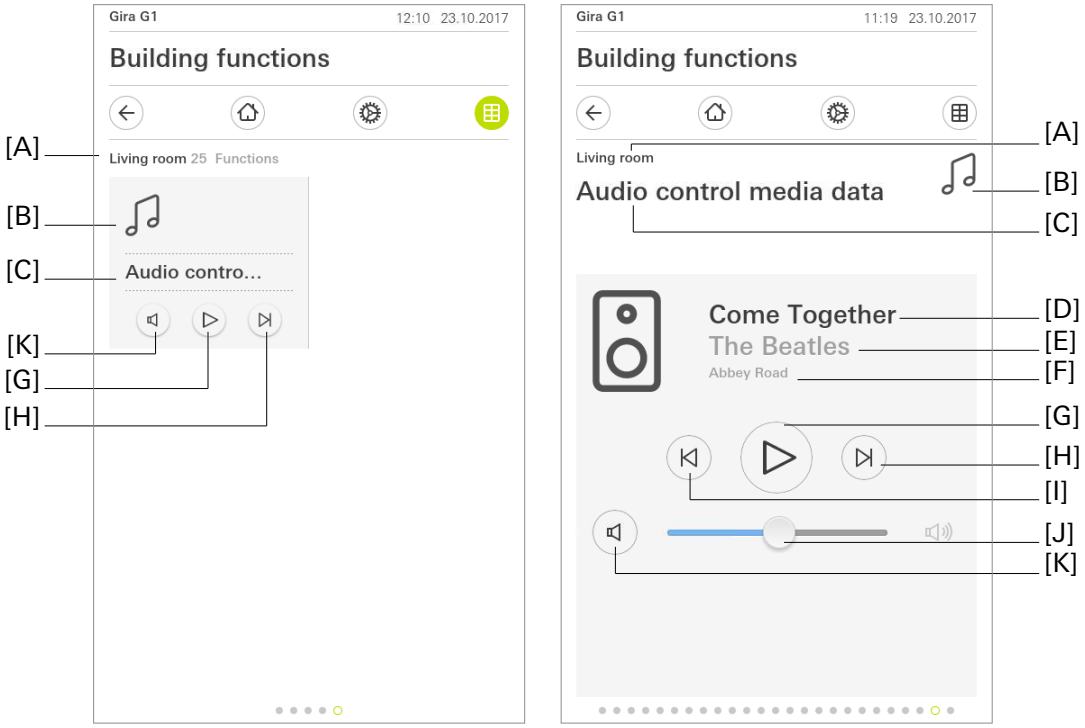


Fig 28: Example of operating and display element of the "Audio control with media data" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of music track
[E] Display of artist
[F] Display of album
[G] Play/Pause
[H] Next track
[I] Previous track
[J] Volume control
[K] Mute button

The Play/Pause button [G] allows you to start or pause playback of a track. The buttons [H] and [I] allow you to change track.

In addition, the volume can be controlled using a slider [J] or switched off completely using the "Mute" button [K]. If information on the track [D], artist [E] and album [F] is available, it is displayed.

The Play/Pause button does not show the current status, but the next possible status. You see the pause icon whilst the music is playing, and vice versa.

### Communication objects

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

A channel with the "Audio control with media data" function has eleven communication objects:

- "Channel n - playback (1-bit):
  1-bit object for starting playback. Polarity: playback active = "1", playback not active = "0"
- "Channel n - feedback playback" (1-bit):
  1-bit feedback object for transmitting the current playback status. Polarity: playback active = "1", playback not active = "0"
- "Channel n - volume" (1-byte):
  1-byte object for setting the volume. Volume between 0% (quiet) and 100% (loud).
- "Channel n - feedback volume" (1-byte):
  1-byte feedback object for reading the current volume. Volume between 0% (quiet) and 100% (loud).
- "Channel n - mute" (1-bit):
  1-bit object for muting. Polarity: muted = "1", not muted = "0".
- "Channel n - feedback muting" (1-bit):
  1-bit feedback object for reading the current muting status.
- "Channel n - previous track" (1-bit):
  1-bit object for playing the previous track.
- "Channel n - next track" (1-bit):
  1-bit object for playing the next track.
- "Channel n - current track" (14-byte):
  14-byte object for displaying the title of the track currently being played
- "Channel n - current album" (14-byte):
  14-byte object for displaying the album name of the track currently being played
- "Channel n - current artist" (14-byte):
  14-byte object for displaying the name of the artist of the track currently being played

### 6.2.24. Audio control with playlist

Using the audio function allows you to control hi-fi systems via KNX. The following functions are available: Play/pause track, change volume, mute, change between tracks (previous and next track), display track, artist, album and playlist and change playlists (previous and next playlist).

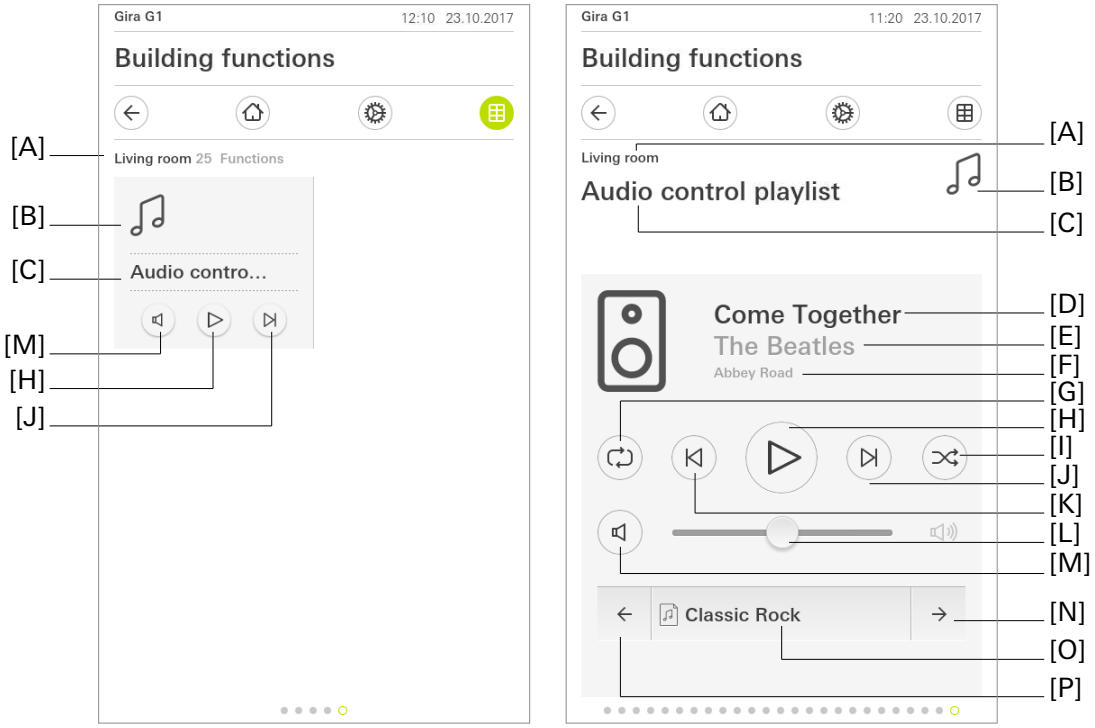


Fig 29: Example of operating and display element of the "Audio control with playlist" function

[A] Display of function folder "User interface description"
[B] Display of "Function icon"
[C] Display of channel function "User interface description"
[D] Display of music track
[E] Display of artist
[F] Display of album
[G] Repeat
[H] Play/Pause
[I] Mix
[J] Next track
[K] Previous track
[L] Volume control
[M] Mute button
[N] Next playlist
[O] Name of current playlist
[P] Previous playlist

The Play/Pause button [G] allows you to start or pause playback of a track. The buttons [J] and [K] allow you to change track. In addition, the volume can be controlled using a slider [L] or switched off completely using the "Mute" button [M]. If information on the track [D], artist [E] and album [F] is available, it is displayed.

The Play/Pause button does not show the current status, but the next possible status. You see the pause icon whilst the music is playing, and vice versa. In addition to the aforementioned functions, the repeat

mode [G] and random playback [I] can be set. Playlists can be changed using the "Next playlist" [N] and "Previous playlist" [P] buttons. The name of the playlist is displayed if available [O].

### Communication objects

A channel with the "Audio control with playlist" function has 18 communication objects:

The names of the communication objects are based on the specifications of the "Description of left structure node" parameter on the "Channel n" parameter page.

- "Channel n - playback (1-bit):
  1-bit object for starting playback. Polarity: playback active = "1", playback not active = "0"
- "Channel n - feedback playback" (1-bit):
  1-bit feedback object for transmitting the current playback status. Polarity: playback active = "1", playback not active = "0"
- "Channel n - volume" (1-byte):
  1-byte object for setting the volume. Volume between 0% (quiet) and 100% (loud).
- "Channel n - feedback volume" (1-byte):
  1-byte feedback object for reading the volume. Volume between 0% (quiet) and 100% (loud).
- "Channel n - mute" (1-bit):
  1-bit object for muting. Polarity: muted = "1", not muted = "0".
- "Channel n - feedback muting" (1-bit):
  1-bit feedback object for reading the current muting status.
- "Channel n - previous track" (1-bit):
  1-bit object for playing the previous track.
- "Channel n - next track" (1-bit):
  1-bit object for playing the next track.
- "Channel n - current track" (14-byte):
  14-byte object for displaying track currently being played
- "Channel n - current album" (14-byte):
  14-byte object for displaying the album name of the track currently being played
- "Channel n - current artist" (14-byte):
  14-byte object for displaying the name of the artist of the track currently being played
- "Channel n - previous playlist" (1-bit):
  1-bit object for changing to the previous playlist
- "Channel n - next playlist" (1-bit):
  1-bit object for changing to the next playlist.
- "Channel n - name of current playlist" (14-byte):
  14-byte object for displaying the name of the current playlist
- "Channel n - mix playlist" (1-bit):
  1-bit object for activating mix mode. Polarity: mix = "1", do not mix = "0".
- "Channel n - feedback mix playlist" (1-bit):
  1-bit feedback object for reading the mix mode. Polarity: mix = "1", do not mix = "0"
- "Channel n - repeat playlist" (1-bit):
  Description: 1-bit object for activating repeat mode. Polarity: repeat = "1", do not repeat = "0".
- "Channel n - feedback repeat playlist" (1-bit):
  Description: 1-bit feedback object for reading the repeat mode. Polarity: repeat = "1", do not repeat = "0".

## 7. Room temperature controller

The Gira G1 can be used for single-room temperature control. Depending on the operating mode, the current temperature setpoint and the room temperature, corrected variables can be emitted to the KNX for heating or cooling control. These corrected variables are generally then evaluated by a suitable KNX actuator, e.g. heating or switching actuators or directly by KNX capable servos and converted to physical variables for room climate control.

The room temperature control is a self-sufficient part of the device. It has its own parameter and object area in the ETS configuration. This means that the room temperature controller can be switched on or off independently of the touch sensor function.

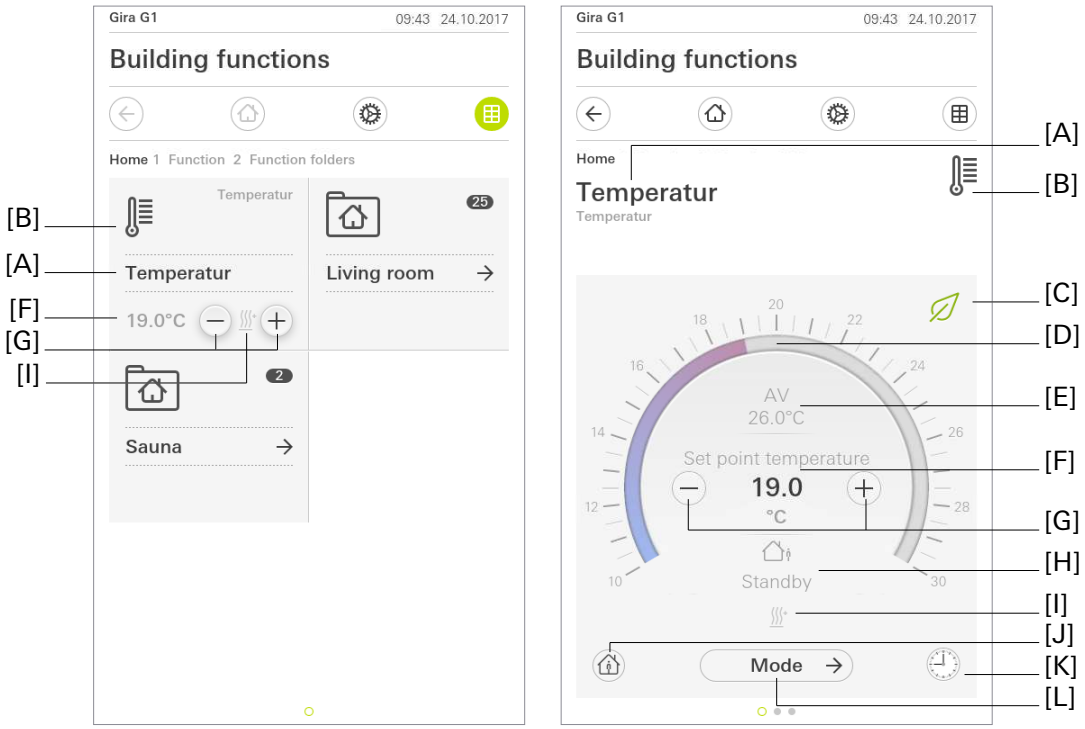


Fig 30: Example of operating and display element of the room temperature controller

[A] Display of channel function "User interface description"
[B] Display of "Function icon"
[C] Display of active eco mode
[D] Operable scale
[E] Display of actual temperature
[F] Display of setpoint temperature
[G] "Value specification" buttons
[H] Display of active operating mode
[I] Display of heating / cooling
[J] "Presence" button
[K] "Timer" button
[L] "Mode" button

The function can be executed in a time-controlled manner using the "Timer" button. The operating modes available are "Comfort", "Standby", "Night" and "Frost/heat protection".

The Gira G1 must not be restarted for 10 minutes after creating or modifying timers, as the changes will not be adopted.

## 7.1. Operating modes and operating mode changeover

### Introduction

The room temperature controller essentially distinguishes between two operating modes. The operating modes define whether the controller should use its corrected variable to control heating systems ("Heating" operating mode) or cooling systems ("Cooling" operating mode). It is also possible to activate mixed operation. In mixed operation, the controller can switch between the "Heating" and "Cooling" modes. Changeover is controlled by the "Heating / cooling toggle" communication object.

The "Operating mode" parameter in the "Room temperature controller (RTC) -> Controller in general" branch of parameters defines the operating mode.

### Individual operating modes of "Heating" or "Cooling"

In the individual operating modes of "Heating" or "Cooling", the controller always works with just one variable in the parameterised operating mode. Depending on the identified room temperature and specified setpoint temperatures of the operating modes, the room temperature controller independently decides whether heating or cooling energy is required and calculates the corrected variable for the heating or cooling system.

### Mixed operating mode of "Heating and cooling"

With the mixed operating mode of "Heating and cooling", the controller can control heating and cooling systems.

The operating mode is switched through the "Heating / cooling toggle" communication object.

The "Heating / cooling toggle" object has the following polarity: "1": heating; "0": cooling. The object value of "0" and the "Operating mode heating/cooling after reset" set in the ETS are activated after a reset. The "Operating mode heating/cooling after reset" parameter allows you to define which operating mode is activated after a reset. With the "Heating" or "Cooling" settings, the controller activates the parameterised operating mode immediately after the initialisation phase. If "Operating mode before reset" is parameterised, the operating mode set before the reset is activated.

In the case of a changeover via the operating mode object, you initially switch to the operating mode specified after reset. Only when the device receives an object update may it switch to the other operating mode.

In addition, the "Heating" and "Cooling" operating mode can be called up and changed via the "Mode" menu.

Setpoint temperature values can be specified in the ETS for each operating mode as part of the initial configuration. The setpoint values for the modes of "Comfort", "Standby" and "Night" can be parameterised directly (absolute setpoint value specification).

Simultaneous heating and cooling (both internal variables calculated as > "0") is not possible. Only in the case of a PWM it is possible for a brief "variable overlap" to occur during the transition between heating and cooling, due to the adjustment of the corrected variable at the end of a time cycle. However, this overlap is amended at the end of a PWM time cycle. Only when heating or cooling energy is required in an operating mode and the variable is therefore > "0" are the icons ⨌⁺ or ₋⨌ shown in the Display.

### Heating / cooling message

Depending on the set operating mode, separate objects can be used to signal whether the controller is currently requesting heating or cooling energy for the control circuit, and therefore whether active heating ⨌⁺ or cooling ₋⨌ is taking place. As long as the corrected variable for heating is > "0“ a "1“ telegram will be transmitted through the "Heating" message object. Only when the corrected variable = "0“ is the message telegram reset ("0“ telegram is transmitted). The same applies to the message object for cooling.

In the case of 2-point control, it should be noted that the icons or are shown in the Display and the message objects for heating or cooling become active as soon as the setpoint temperature value of the active operating mode is fallen short of for heating or exceeded for cooling. The parameterised hysteresis is not taken into account!

The message objects can be enabled through the "Heating message“ and "Cooling message" parameters in the "Room temperature controller (RTC) -> Variable and status output" branch of parameters. The control algorithm controls the message objects. The corrected variable is calculated, and therefore the message objects are updated, every 30 seconds only.

### 7.2. Control algorithms and corrected variable calculation

#### Introduction

For convenient temperature control in a residential or commercial room, a special control algorithm is required which controls the installed heating or cooling systems. With it the controller determines corrected variables which control the heating or cooling system while taking the setpoint temperature specifications and the actual room temperature into account. The control system (control circuit) consists of the room temperature controller, the servo or the switching actuator (when using electrothermal drives (ETDs)), the actual heating or cooling element (e.g. radiator or cooling ceiling) and the room. This results in a regulated system (Fig. 31).

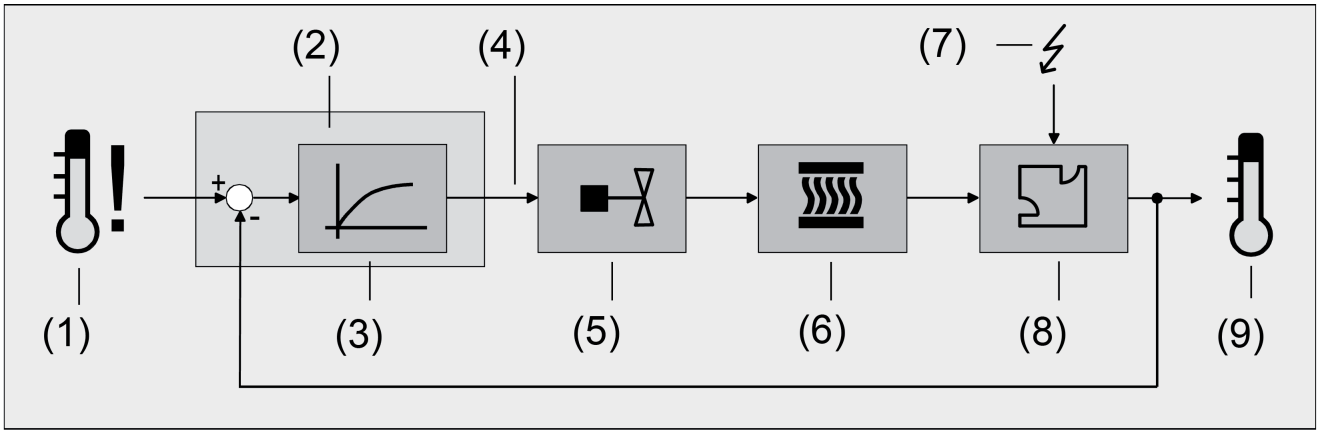


Fig 31: Regulated system of a single-room temperature control

(1) Setpoint temperature specification
(2) Room temperature controller
(3) Control algorithm
(4) Variable
(5) Valve control (servo, ETD, heating actuator, etc.)
(6) Heat/cold exchanger (radiator, cooling ceiling, FanCoil, etc.)
(7) Disturbance variable (sunlight, outdoor temperature, lighting systems, etc.)
(8) Room
(9) Actual temperature (room temperature)

The controller works with the actual temperature (9) from the room temperature measurement of the Gira G1 and compares it to the specified setpoint temperature (1).

The corrected variable (4) is calculated from the difference between the actual and the setpoint temperature using the

control algorithm (3). Valves or blowers for heating or cooling systems are controlled with the corrected variable (5), causing heating or cooling energy in the heat or cold exchangers (6) to be emitted to the room (8).

Regular readjustment of the corrected variable enables the controller to compensate setpoint/actual temperature differences caused by external influences (7). In addition, the flow temperature of the heating or cooling circuit acts on the control systems, making corrected variable adjustments necessary.

The room temperature controller allows you to choose between constant or switching proportional/integral control (PI) and switching 2-point control.

The internal controller of the Gira G1 can be run in three different operating modes. The operating mode can be parameterised to "Heating", "Cooling" or "Heating and cooling".

The corrected variables calculated by the control algorithm are output through the "Heating variable" or "Cooling variable" communication objects. The format of the corrected variable objects is defined depending on the control algorithm selected for heating and/or cooling mode. This means that corrected variable objects of 1 bit or 1 byte in size can be created. The control algorithm is defined by the "Type of heating control" or "Type of cooling control" parameters in the "Room temperature controller (RTC) -> Controller in general" branch of parameters.

**Constant PI control**

A PI control is understood to be an algorithm which consists of a proportional and an integral part. Combining these control properties makes the fastest and most accurate control of the room temperature possible without or with only minimal control deviations.

With this algorithm, the room temperature controller cyclically calculates a new constant corrected variable every 30 seconds and outputs these to the KNX with a 1-byte value object if the calculated corrected variable value has changed by a defined percentage. The "Automatic transmission after change of" parameter in the "Room temperature controller (RTC) –> Variable and status output" branch of parameters determines the change interval in percent.

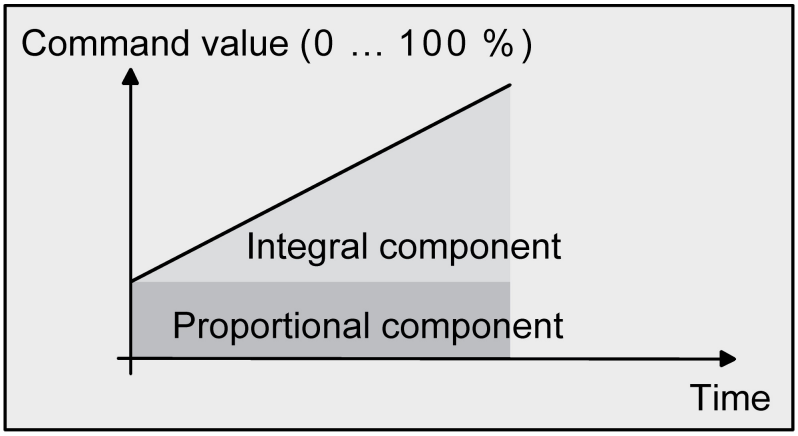


Fig 32: Constant PI control

**Switching PI control**

Also with this type of control, the room temperature is kept constant through the PI control algorithm. Averaged over time, this results in the same control system behaviour as with a constant controller. It differs from constant control only in terms of variable output. The corrected variable calculated cyclically by the algorithm cyclically every 30 seconds is internally converted to an equivalent pulse width-modulated (PWM) variable signal and emitted to the KNX via a 1-bit switching object after the end of the cycle time. Taking into account the cycle time, which can be set through the "Cycle time for the switching variable" parameter in the "Room temperature controller (RTC) –> Variable and status output" branch of parameters, the average of the corrected variable signal resulting from this modulation is a measurement of the average position of the control valve, and therefore a reference for the set room temperature.

A shift in the average, and therefore a change in the heating power, is achieved by modifying the duty cycle of the switch-on and switch-off pulse of the variable signal. The duty cycle is only adjusted by the controller at the end of a time period based on the calculated corrected variable! Every variable change

is implemented, regardless of the amount by which the variable changes (the parameters "Automatic transmission after change of" and "Cycle time for automatic sending" have no function here).

The last corrected variable value calculated in an active time period is implemented. The variable is also only adjusted at the end of an active cycle time in the case of a change to the setpoint temperature, for example through a change in the operating mode. The following image shows the emitted variable switching signal based on the internally calculated corrected variable value (initially 30%, then 50% variable; variable output not inverted).

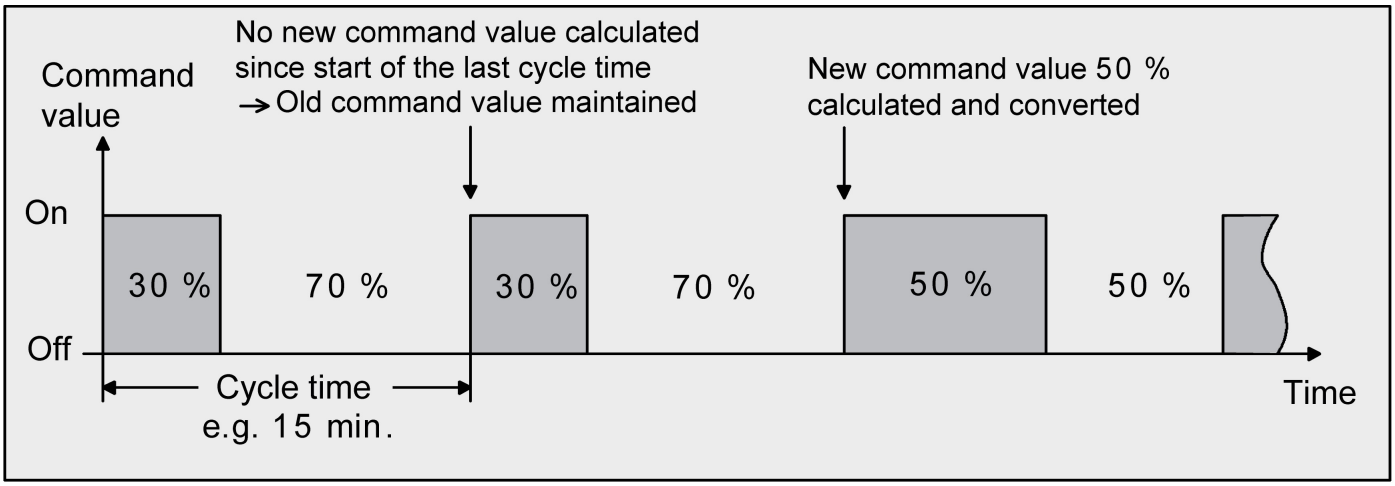


Fig 33: Switching PI control

In the case of a variable of 0% (permanently switched off) or 100% (permanently switched on), a corrected variable telegram corresponding to the corrected variable value ("0" or "1") is always emitted at the end of a cycle time.

The controller always calculates internally with constant corrected variable values in the case of switching PI control. These constant values can also be emitted to the KNX via a separate 1-byte value object, for example for visualisation as status information. The status value objects are only updated after the end of the parameterised cycle time, along with the variable output. The "Automatic transmission after change of" and "Cycle time for automatic sending" parameters have no function here. All PWM controls use the same cycle time.

**Cycle time**

Pulse width-modulated corrected variables are primarily used to control electrothermal drives (ETDs). The room temperature controller transmits the switching corrected variable telegrams to a switching actuator with a semiconductor switching element, to which the drives are connected (e.g. heating actuator or room actuator). Setting the cycle time of the PWM signal on the controller makes it possible to adjust the control to suit the drives being used. The cycle time defines the switching frequency of the pulse width-modulated signal and permits adjustment to the adjustment cycle times of the servos being used (travel time that the drive requires to adjust the value from fully closed position to fully open position). In addition to the adjustment cycle time, the dead time (time during which the servos show no reaction when switching on or off) must be taken into account. If different drives with different adjustment cycle times are used, the greater of the times should be taken into account. Remember to consider the instructions of the drive manufacturer.

It is essentially possible to distinguish between two cases when configuring cycle time:

**Case 1: Cycle time > 2 x adjustment cycle time of the electrothermal drives (ETDs) used**

In this case, the switch-on or switch-off times of the PWM signal are so long that the drives have sufficient time to fully open or close during a time period.

Advantages:

The desired variable average, and therefore the requested room temperature, is set relatively precisely, even when several drives are being controlled at the same time.

Disadvantages:
It should be noted that the drives' expected service life can be reduced by constantly implementing the full valve travel. In some cases, if the cycle times are very long (> 15 minutes) and the system is less sluggish, the heat dissipation into the room around the radiator may be uneven and perceived as problematic.

This cycle time setting is recommended for sluggish heating systems (e.g. underfloor heating).

This setting is also recommended if you have a larger number of controlled drives, perhaps different drives, so that the travel of the valves can be averaged better.

**Case 2: Cycle time < adjustment cycle time of the electrothermal drives (ETDs) used**

In this case, the switch-on or switch-off times of the PWM signal are so short that the drives do not have sufficient time to fully open or close during a period.

Advantages:
This setting ensures a continual flow of water through the radiators, therefore allowing for even heat dissipation into the room.

If just one servo is being controlled, the controller is able to compensate for the average shift caused by the short cycle time by continually adjusting the corrected variable, therefore setting the desired room temperature.

Disadvantages:
If more than one drive is being controlled at the same time, the desired average for the corrected variable, and therefore the desired room temperature, is difficult to set, or has considerable deviations.

The continuous flow of water through the value and the constant warming of the drive causes a change in the dead times of the drives during the opening and closing phase. Taking into account the dead times, the short cycle time means that in some cases the requested corrected variable (average) can only be set with a considerable deviation. So that the room temperature can be constantly adjusted after a certain amount of time, the controller must compensate for the average shift caused by the short cycle time by continually adjusting the corrected variable. Normally the control algorithm (PI control) in the controller takes care of balancing out control deviations.

This cycle time setting is recommended for fast-acting heating systems (e.g. panel heaters).

**2-point control**

2-point control is a very simple type of temperature control. With this type of control, two hysteresis temperature values are specified. The actuators are controlled by means of switch-on and switch-off corrected variable commands (1 bit) from the controller. A constant corrected variable is not calculated with this type of control.

Also with this type of control, the room temperature is cyclically evaluated every 30 seconds. This means that the corrected variables only change at these times, if required. The advantage of the very simple 2-point room temperature control contrasts with the constantly fluctuating temperature given by this type of control. For this reason, fast-acting heating or cooling systems should not be controlled using 2-point control as this can cause major swings in temperature, and therefore a loss of convenience and comfort. A difference should be drawn between the operating modes when defining the hysteresis limits.

**Individual operating modes of "Heating" or "Cooling"**

In heating mode, the controller switches the heating on when the room temperature falls below a set limit. In heating mode, the controller only switches the heating off again when a set temperature limit is exceeded.

In cooling mode, the controller switches the cooling on when the room temperature rises above a set limit. The cooling is only switched off again when the temperature drops below a set limit. Depending on the switching status, variable "1" or "0" is emitted if the temperature exceeds or falls below the hysteresis limits.

The hysteresis limits of both operating modes can be configured in the ETS.

The following images show 2-point control for the individual operating modes of "Heating" (Fig. 34) or "Cooling" (Fig. 35). The images involve two setpoint temperature values, single-stage heating or cooling and non-inverted variable output.

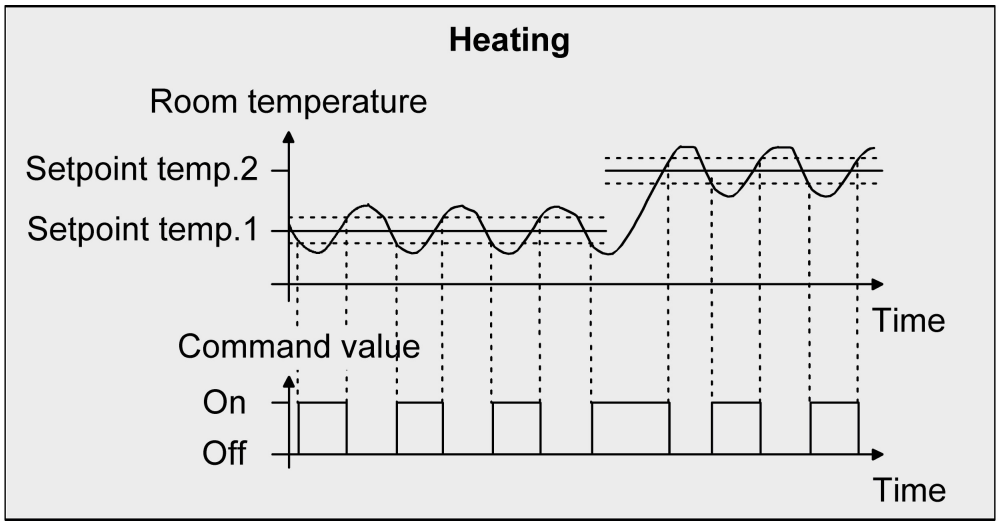


Fig 34: 2-point control for the "Heating" operating mode

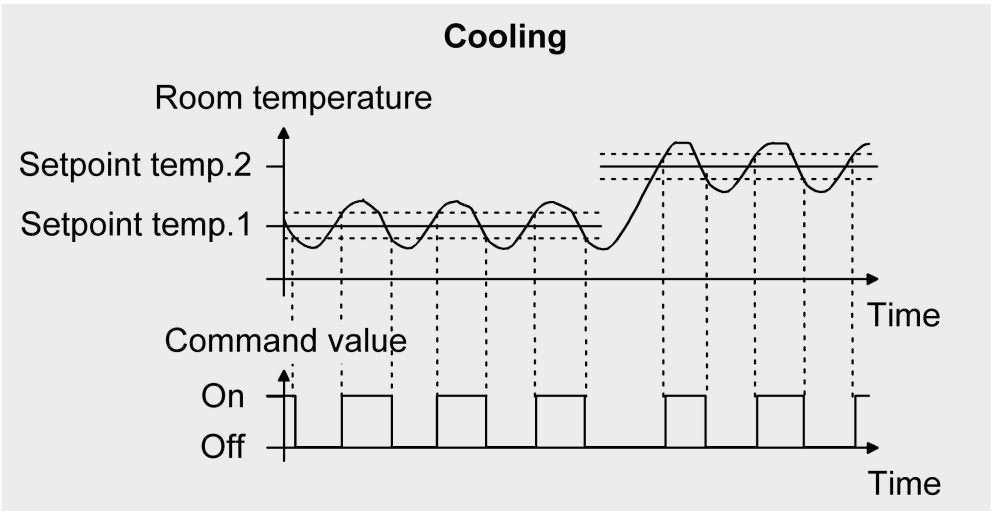


Fig 35: 2-point control for the "Cooling" operating mode

**Mixed operating mode of "Heating and cooling":**

In the event of operating mode changeover through the object, in heating mode the controller switches the heating on when the room temperature falls below a set hysteresis limit. In heating mode, the controller only switches the heating off again when the set upper hysteresis limit is exceeded. Similarly, in cooling mode, the controller switches the cooling on when the room temperature rises above a set hysteresis limit.

In cooling mode, the controller only switches the cooling off again when the set lower hysteresis limit is fallen short of. As with the individual operating modes of heating or cooling, two hysteresis limits exist per operating mode. There is also a dead zone for calculating the setpoint temperature values for cooling, but the dead zone does not affect the calculation of the 2-point variable as changeover of operating mode only takes place manually by means of the corresponding object. This means that within the hystereses it is possible for heating or cooling energy to still be requested, even in the case of temperature values that are in the dead zone.

The following two images show 2-point control for the mixed operating mode of "Heating and Cooling", split into heating mode (Fig. 36) and cooling mode (Fig. 37). The images involve two setpoint temperature values, non-inverted variable output and automatic operating mode changeover. In the case of operating mode changeover through the object, it is also possible to parameterise an upper hysteresis for heating and a lower hysteresis for cooling.

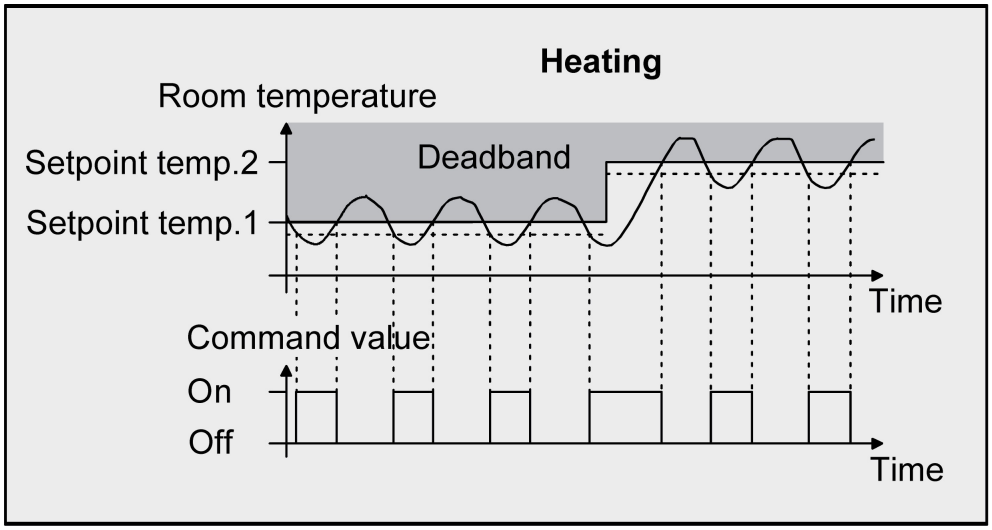


Fig 36: 2-point control for the mixed operating mode of "Heating and cooling" in active heating mode

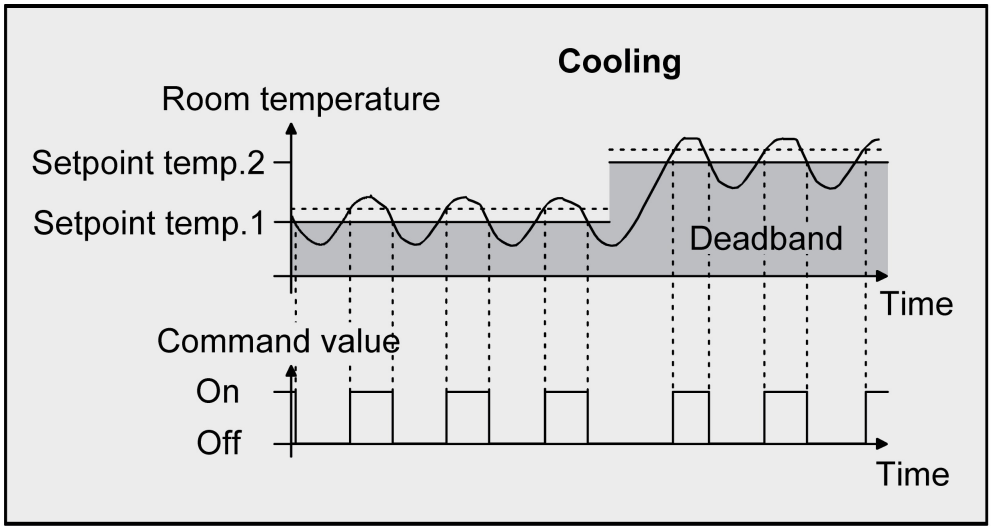


Fig 37: 2-point control for the mixed operating mode of "Heating and cooling" in active cooling mode

Depending on the switching status, variable "1" or "0" is emitted if the temperature exceeds or falls below the hysteresis limits or setpoint values.

## 7.3. Adjustment of control algorithms

### Adjustment of PI control

There are various systems which can heat up or cool down a room. For example, it is possible to heat or cool the environment uniformly with heat carriers (preferably water or oil) in conjunction with a room air convection system. Such systems are used, for example, for wall radiators, floor heating systems or cooling ceilings.

Blower systems can heat or cool rooms as an alternative or additionally. Such systems are usually equipped with electric blower heaters, blower cooling systems or cooling compressors with blowers. Such heating or cooling systems are usually quite fast, as they heat up the room air directly.

To enable the PI control algorithm to efficiently control all common heating or cooling systems, and therefore the room temperature control to function as quickly as possible without control deviation, it is necessary to calibrate the control parameters. With PI control certain factors can be set for this purpose which have a major effect on the control behaviour. For this reason, the room temperature controller can be set to predefined values based on experience for the most common heating or cooling systems. If no satisfactory control results can be achieved with preset values by selecting a corresponding heating or cooling system, it may be possible to optimise the adjustment with control parameters.

The "Type of heating" or "Type of cooling" parameters are used to set predefined control parameters for the heating or cooling level. These fixed values correspond to practical values of a properly planned and executed air conditioning system and result in an optimal behaviour of the temperature control. The types of heating or cooling shown in the following tables can be defined for heating or cooling mode.

| Heating type | Proportional range (preset) | Readjustment time (preset) | Recommended PI control type | Recommended PWM cycle time |
|---|---|---|---|---|
| Hot-water heating | 5 kelvin | 150 minutes | constant/PWM | 15 min. |
| Floor heating | 5 kelvin | 240 minutes | PWM | 15-20 min. |
| Electric heating | 4 kelvin | 100 minutes | PWM | 10-15 min. |
| Fan coil unit | 4 kelvin | 90 minutes | constant | --- |
| Split unit (divided air conditioner) | 4 kelvin | 90 minutes | PWM | 10-15 min. |

Table 1:Predefined control parameters and recommended control types for heating systems

| Cooling type | Proportional range (preset) | Readjustment time (preset) | Recommended PI control type | Recommended PWM cycle time |
|---|---|---|---|---|
| Cooling ceiling | 5 kelvin | 240 minutes | PWM | 15-20 min. |
| Fan coil unit | 4 kelvin | 90 minutes | constant | --- |
| Split unit (divided air conditioner) | 4 kelvin | 90 minutes | PWM | 10-15 min. |

Table 2:Predefined control parameters and recommended control types for cooling systems

If the "Type of heating" or "Type of cooling" parameters are set to "via control parameters", it is possible to adjust the control parameters. Control can be decisively influenced by specifying the proportional range for heating or cooling (P-percentage) and the readjustment time for heating or for cooling (I-percentage).

Changing a control parameter by even small values will result in a considerably different control behaviour!

The starting point for the adjustment should be the control parameter setting of the corresponding heating or cooling system in accordance with the fixed values specified in Table 1 and Table 2.

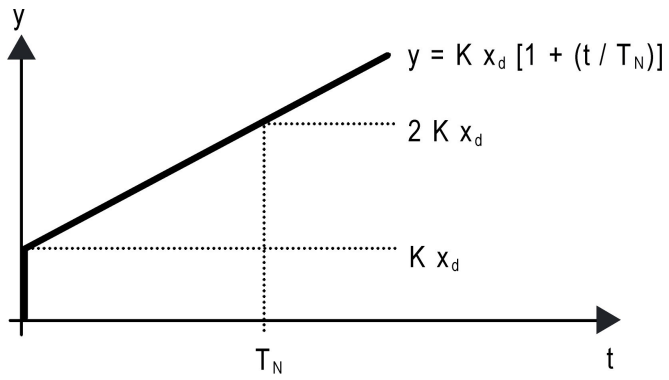


Fig 38: Function of the corrected variable of a PI control

y: Variable
$x_d$: Control difference ($x_d = x_{setpoint} - x_{actual}$)
P = 1/K : parameterisable proportional range K = 1/P : amplification factor
$T_N$: parameterisable readjustment time

PI control algorithm: Corrected variable $y = K x_d [1 + (t / T_N)]$
Upon deactivation of readjustment time (setting = "0") -> P control algorithm: Corrected variable $y = K x_d$

| Parameter setting | Effect |
|---|---|
| P: small proportional range | large overshoot with setpoint value changes (possibly also continuous oscillation), fast adjustment to the setpoint value |
| P: large proportional range | no (or small) overshoot, however slow adjustment |
| $T_N$: short readjustment time | fast correction of control deviations (ambient conditions), danger of continuous oscillation |
| $T_N$: long readjustment time | slow correction of control deviations |

Table 3:Effects of settings for control parameters

### Adjustment of the 2-point control

2-point control is a very simple type of temperature control. With this type of control, two hysteresis temperature values are specified. The upper and lower temperature hysteresis limit can be set through parameters. It should be remembered that:

- a low hysteresis leads to lower temperature fluctuations but a higher bus load,
- a high hysteresis switches less often, but causes uncomfortable fluctuations in temperature.

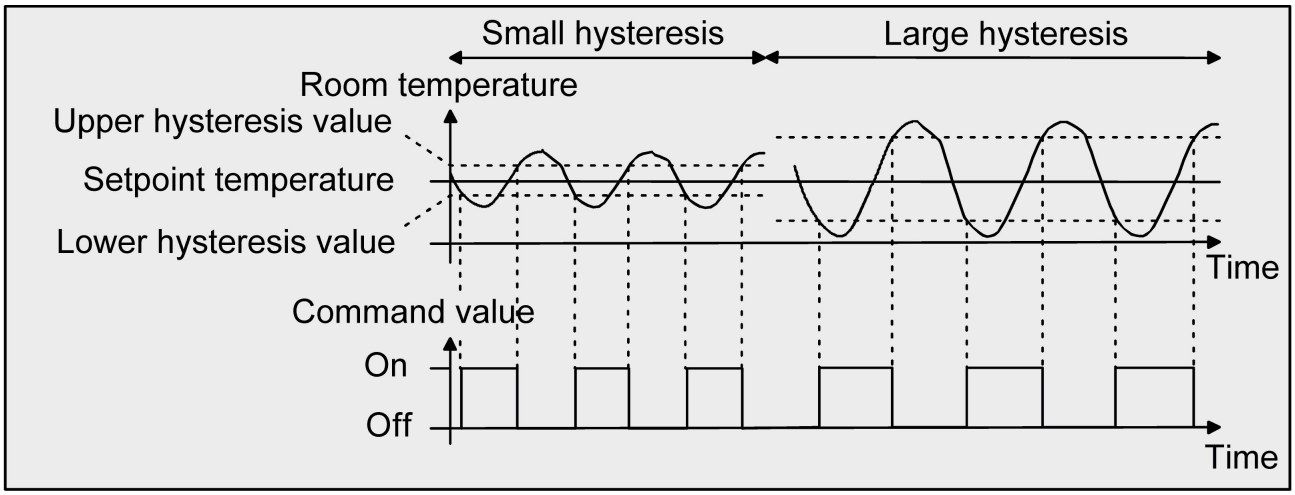


Fig 39: Effects of the hysteresis on the switching behaviour of the variable of a 2-point control

### 7.4. Operating mode changeover

**Introduction - The operating modes**

The room temperature controller differentiates between various operating modes. For example, by activating these modes it is possible to activate various setpoint temperature values, depending on the presence of a person, the condition of the heating or cooling system, the time of day or the day of the week. The following operating modes are differentiated:

- Comfort mode
  Comfort mode is generally activated if people are in a room and therefore the room temperature is to be set to a comfortable, reasonable value. The changeover to this operating mode can be carried out by tapping on the "Mode" button, through the "Operating mode changeover" communication object or presence-controlled, for example by a PIR sentinel on the wall or a presence detector on the ceiling.
  If comfort mode is activated, this is indicated in the Display by means of the ⌂ icon.
- Standby mode
  When a room is not used during the day, as persons are not present, standby mode can be activated. As a result, the room temperature can be adjusted to a standby value, enabling heating or cooling energy to be saved.
  If standby mode is activated, this is indicated in the Display by means of the ⌂ icon.
- Night mode
  During night hours or a longer absence, it usually makes sense to adjust the room temperature to cooler temperatures (e.g. in bedrooms). In this case, cooling systems can be set to higher temperature values when air conditioning is not necessary (e.g. in offices). The night mode can be activated for this purpose.
  If night mode is activated, this is indicated in the Display by means of the ☾ icon.
- Frost/heat protection mode
  Frost protection is required when, for example, the room temperature is not to fall below critical values when a window is open. Heat protection can be required when the temperature in an environment which is usually warm due to external influences becomes too high. In these cases, freezing or overheating of the room can be prevented by specifying an individual temperature setpoint by activation of the the frost/heat protection, depending on the "Heating" or "Cooling" operating mode.
  If frost/heat protection is activated, this is indicated in the Display by means of the ❄ icon.
- Comfort extension (temporary comfort mode)
  The comfort extension must be activated from within night mode or frost/heat protection mode (not triggered by the "Window status" object) and can be used to adjust the room to the comfort temperature for a certain time. This may, for example, be the case when the room is also used during the night. Activation is only carried out by the "Presence" button or by the presence object. The comfort extension

is automatically deactivated after a specifiable time has expired, when the "Presence" button is pressed again or by receiving a presence object value = "0". The extension cannot be re-triggered.
If a comfort extension is activated, this is indicated in the Display by means of the green "Presence" button.

An independent setpoint temperature value can be specified for each operating mode of the "Heating" or "Cooling" operating types.

Only one operating mode can be active.

**Operating mode changeover**

The operating modes can be activated or changed over in various ways. An activation or changeover is – interdependent according to priorities – possible by:

- on-site operation of the room temperature controller and parameterised operating mode changeover,
- the KNX objects (as well as a controller extension).

The options for operating mode changeover are described in detail in the following.

**Changeover of operating mode in the mode menu**

If the room temperature controller function is switched on, the room temperature controller is shown in a tile on the Home screen with the configured function icon and entered description of the user interface. An operating mode changeover can only be carried out in the detail view on the Gira G1. You can jump to the detail view using the ⊞ button or by tapping on the appropriate tile. In the detail view, the menu for changing the operating mode is called up using the "Mode ->" button. The four operating modes of comfort, standby, night and frost/heat protection and the corresponding setpoint temperatures are listed on the "Operating mode" page. Depending on the parameterised operating mode, the setpoint temperature for heating, cooling or heating and cooling is shown in the Display. Tapping on an operating mode selects it. The selected operating mode is activated using the ok button. In addition, the setpoint temperatures of the various operating types can be adjusted in the different operating modes. Tapping on the "Edit" button opens up the "Edit operating mode" page, on which you can select the operating mode for which you would like to adjust the setpoint value. Swipe vertically with your finger to individually adjust the setpoint temperature. With the exception of the frost/heat protection, the setpoint temperatures can be edited for all operating modes. Because the frost/heat protection mode is a safety function, it is only possible to adjust these setpoint temperatures through the parameters in the ETS.

The presence detection, window status and forced object for operating mode changeover have higher priority than changeover of the operating mode on site on the "Operating mode" page. Changeovers through evaluation of the corresponding objects therefore take precedence.

If night mode or frost/heat protection is activated (not activated by the "Window status" object), the "Presence" button can be used to switch to the comfort extension or temporarily deactivate it again. The "Presence" button can also be used to switch to comfort mode from standby mode.

**Changeover of operating mode through KNX communication objects**

The "Operating mode changeover" parameter in the "Room temperature controller (RTC) -> Controller in general" branch of parameters defines the changeover procedure as follows:

- Operating mode changeover "with value (1-byte)"
  A common 1-byte changeover object exists for all operating modes. The operating mode can be changed over immediately after receiving just one telegram during the running time with this value object. The received value specifies the operating mode for this purpose. In addition, a second 1-byte object is available which can set an operating mode with forced control and priority, regardless of all other changeover options. Both 1-byte objects are implemented in accordance with the KNX specification. While taking the priority into account, a certain changeover hierarchy results during an operating mode changeover by the objects. Here a distinction is made between presence detection with a presence button (Fig. 40) or a presence detector (Fig. 41). In addition, the status of the window in the room can be evaluated with the "Window status" object. This enables the controller to change to the frost/heat protection mode for saving energy with the window opened, regardless of the primarily set operating mode.
  Table 4, on page 117 also shows the states of the communication objects and the resulting operating mode.

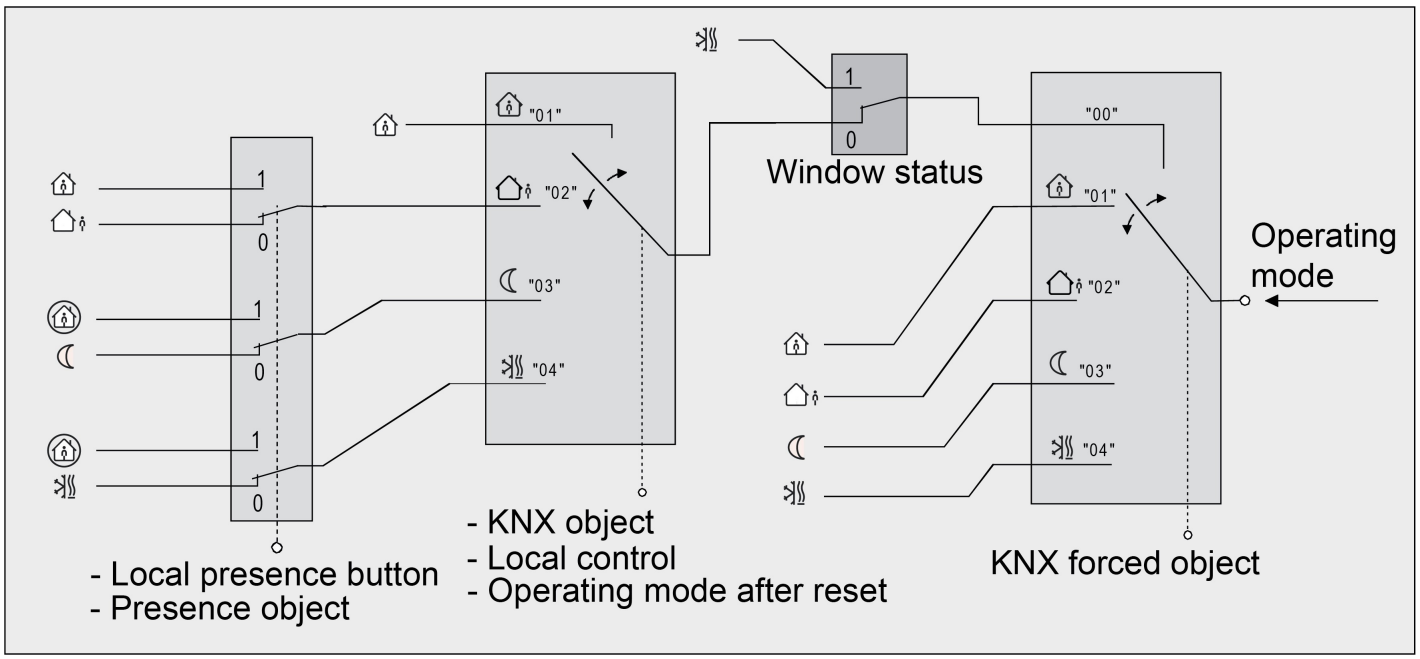


Fig 40: Operating mode changeover by KNX object with presence button

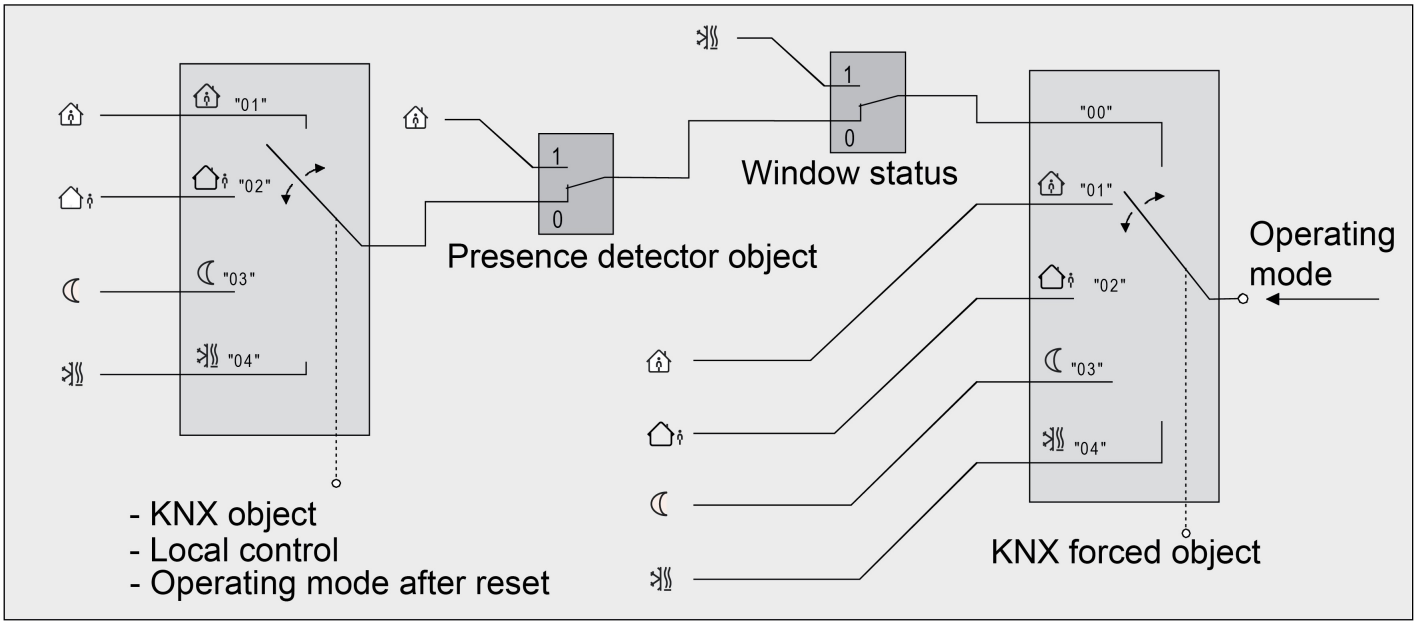


Fig 41: Operating mode changeover by KNX object with presence detector

| "Operating mode changeover" object value | "Forced object operating mode" object value | "Window status" object | "Presence" button | Presence detector | Resulting operating mode |
|---|---|---|---|---|---|
| 00 | 00 | 0 | X | 0 | No change |
| 01 | 00 | 0 | 0 | - | Comfort mode |
| 02 | 00 | 0 | 0 | - | Standby mode |
| 03 | 00 | 0 | 0 | - | Night mode |
| 04 | 00 | 0 | 0 | - | Frost/heat protection |
| 01 | 00 | 0 | 1 | - | Comfort mode |
| 02 | 00 | 0 | 1 | - | Comfort mode |
| 03 | 00 | 0 | 1 | - | Comfort extension |
| 04 | 00 | 0 | 1 | - | Comfort extension |
| 01 | 00 | 0 | - | 0 | Comfort mode |
| 02 | 00 | 0 | - | 0 | Standby mode |
| 03 | 00 | 0 | - | 0 | Night mode |
| 04 | 00 | 0 | - | 0 | Frost/heat protection |
| X | 00 | 0 | - | 1 | Comfort mode |
| X | 00 | 1 | - | X | Frost/heat protection |
| X | 00 | 1 | X | - | Frost/heat protection |
| X | 01 | X | X | X | Comfort mode |
| X | 02 | X | X | X | Standby mode |
| X | 03 | X | X | X | Night mode |
| X | 04 | X | X | X | Frost/heat protection |

Table 4:States of the communication objects and the resulting operating mode

X: State irrelevant

-: Not possible

When switching an operating mode, for example by means of on-site control, the KNX changeover object is updated by the controller and can be read out if the "Read" flag is in place. If the "Transmit" flag is in place for this object, the current value is also automatically sent out to the bus upon modification.

After a device reset, the value corresponding to the set operating mode is sent to the bus if the "Transmit" flag is in place.

Changeover through the KNX object "Operating mode changeover" has equal value to changeover on site on the device. An operating mode specified by the object (e.g. through a controller extension) can be adjusted by means of operating mode changeover on the device if no higher-priority mode (e.g. window contact / presence detector) or the KNX forced object is activated.

The KNX forced object always has top priority.

"Presence" button: The presence object is active for the duration of an activated comfort extension ("1"). The presence object is automatically deleted ("0") when the comfort extension is ended after the extension time expires, the operating mode has been changed by operation by the changeover objects or by on-site operation, or a forced operating mode is deactivated by the KNX forced object (forced object -> "00"). The controller automatically resets the state of the "Presence" button when an object value is received via the operating mode object or the forced object is reset.

**Further information on the presence function / comfort extension**

With presence detection, the room temperature controller can be briefly switched into the comfort extension by tapping on the "Presence" button or switched to comfort mode in case of movement by persons present in the room. Here, the "Presence detection" parameter in the "Room temperature controller (RTC) -> Controller functionality" parameter node defines whether the presence detection takes place through movement using a presence detector or manually by tapping on the "Presence" button on the Display:

- Presence detection with presence button
  The "Presence object" is enabled. This means that if night mode or frost/heat protection is activated (not activated by the "Window status" object), a press of the "Presence" button or receipt of a presence object value = "1" causes a switch to the comfort extension. The extension is automatically deactivated as soon as the parameterised "Duration of comfort extension" has expired. A comfort extension can be deactivated prematurely if the "Presence" button is pressed again or a value = "0" is received via the presence object. Retriggering of the extension time is not possible.
  If the "Duration of comfort extension" is set to "0" in the ETS, a comfort extension cannot be activated from night mode or frost/heat protection mode. In this case, the operating mode is not changed, although the presence function is activated.
  If the standby mode is active, the system can be switched into the comfort mode by a presence object value = "1". This also happens if the duration of the comfort extension is set to "0". Comfort mode remains active in the process for as long as the presence function remains active or until another operating mode is set.
  The presence object or the presence function is always deleted during a changeover to another operating mode or after deactivation of a forced operating mode (with KNX forced changeover). A presence function activated before a device reset (programming process, bus voltage failure) is always deleted after the reset, including the object value.
- Presence detection by presence detector
  If a presence detector is configured as a presence detection device, the controller only evaluates the "Presence object". Presence detectors can be integrated into room temperature control with this object. If a movement is detected ("1" telegram), the controller switches into the comfort mode. Here, specifications made by changeover objects or on-site operation directly on the device are not relevant. Only a window contact or the KNX forced object has higher priority. After the movement delay time in the presence detector expires ("0" telegram), the controller switches back into the mode active before the presence detection or it tracks the telegrams of the operating mode objects received during the presence detection. Changeover of the operating mode on the room temperature controller during active presence detection is not possible.

A presence function activated before a device reset (programming process, bus voltage failure) is always deleted after the reset, including the object value. In this case, the presence detector must send a new "1" telegram to the controller to activate the presence function.

### Further information on the window status

The room temperature controller is equipped with various options for switching into the frost/heat protection mode. In addition to changeover through the corresponding operating mode changeover object or by room temperature controller operation on the device, the frost/heat protection mode can be activated by a window contact. The window contact is assigned a higher priority with these options.

The 1-bit "Window status" object is enabled. A telegram with the value = "1" (opened window) to this object activates the frost/heat protection mode. If this is the case, the operating mode cannot be deactivated by either on-site operation or the changeover objects (with the exception of the KNX forced object) or heating timer. The window status is not reset and the frost/heat protection mode is not deactivated until a telegram with the value = "0" (closed window) is received. Then the operating mode set before the window was opened or the operating mode updated via the KNX or heating timer while the window was opened is activated.

It is possible to parameterise a delay in the window status if desired. This delay may be useful if you merely wish to ventilate the room by opening the window and do not wish to cause a change in operating mode. The delay time is set through the "Delay window status" parameter and can be between 1 and 255 minutes. Only after the parameterised time is the window status, and therefore the frost/heat protection, activated. A setting of "0" causes immediate activation of the frost/heat protection when a window is open. The window status is effective in the heating and in the cooling mode. The value of the "Window status" object is deleted after a reset.

When a window is open, the controller operating mode cannot be switched through the "Operating mode changeover" communication object or on-site operation on the device. The operating mode is changed over after the window is closed.

### Further information on the operating mode after a reset

The "Operating mode after reset" parameter under the "Room temperature controller (RTC) -> Controller in general" parameter node in the ETS allows you to specify which operating mode is to be activated after bus voltage is restored or after programming through the ETS. The following settings are possible:

- "Comfort mode" -> Comfort mode is activated after the initialisation phase.
- "Standby mode" -> Standby mode is activated after the initialisation phase.
- "Night mode" -> Night mode is activated after the initialisation phase.
- "Frost/heat protection mode" -> Frost/heat protection mode is activated after the initialisation phase.
- "Restore operating mode before reset" -> The mode set before a reset according to the operating mode object, heating timer or mode menu of the room temperature controller (normal priority) is set after the device's initialisation phase.
  Operating modes set before the reset using a function with a higher priority (forced, window status, presence status) are not updated.

The operating mode objects are updated after a reset.

## 7.5. Setpoint temperature values

### Setpoint temperature specification

Setpoint temperature values can be specified in the ETS for each operating mode as part of the initial configuration. The setpoint values for the modes of "Comfort", "Standby" and "Night" can be parameterised directly (absolute setpoint value specification). If desired, setpoint temperatures can be adjusted later during ongoing operation by operating the controller on site or using KNX communication objects.

For the "Frost/heat protection" operating mode, two separate setpoint temperature values for heating mode (frost protection) and cooling mode (heat protection) can only be configured in the ETS. These temperature values cannot be adjusted later, during operation of the controller.

The "Setpoint value specification" on the "Room temperature controller (RTC) -> Setpoint values" parameter page always takes place using absolute temperature values.

The setpoint temperatures for the comfort, standby and night mode are independent of each other. Various temperature values in the range from

+7.0 °C to +40.0 °C can be specified for each operating mode and operating type in the ETS. The ETS does not validate the temperature value. For example, it is possible to specify lower setpoint temperatures for the cooling mode than for the heating mode or lower temperatures for the comfort mode than for the standby mode.

After start-up through the ETS, the setpoint temperatures can be changed via the bus by temperature telegrams. The "Setpoint value active operating mode" communication object is available for this purpose. If the controller receives a telegram via this object, it immediately sets the received temperature as a new setpoint value of the active operating mode and continues to work with this setpoint value. This makes it possible for the setpoint temperatures of all operating modes to be adjusted separately for heating and cooling mode. The frost or heat protection temperature programmed through the ETS cannot be changed in this way.

The controller rounds the temperature values received through the object to the specified step width of the setpoint offset. The step width can be configured in the ETS (0.1 K / 0.2 K / 0.5 K / 1.0 K).

The setpoint temperature values programmed into the room temperature controller upon start-up through the ETS can be modified during operation of the device on site using the room temperature controller's mode menu or communication objects. In the ETS, the "Overwrite user data during an ETS programming procedure?" parameter on the "Room temperature controller (RTC) -> Controller in general" parameter page allows you to specify whether the setpoint values in the device, which may have been subsequently modified, should be overwritten and replaced with the values parameterised in the ETS during an ETS programming procedure. If the checkmark is placed beside this parameter, the setpoint temperature values on the device are deleted and replaced with the values from the ETS during a programming procedure. If the checkmark is not placed, the setpoint values in the device remain unchanged. The setpoint temperatures entered in the ETS then have no significance.

Upon initial start-up of the device, the checkmark beside the "Overwrite user data during an ETS programming procedure?" parameter must be placed in order to properly initialise the memory locations in the device. This setting is also necessary if key controller properties (operating mode, setpoint value specification, etc.) are modified with new parameter configurations in the ETS!

### Applying setpoint values

Changes are applied temporarily in the case of modification of setpoint temperatures by means of the "Setpoint value active operating mode" communication object or adjustment in the mode menu of the room temperature controller.

- The setpoint value change is only applied temporarily:
  The setpoint values set on the room temperature controller or received by the objects only remain active temporarily. In the case of a bus voltage failure, after a changeover of the operating mode (e.g.

comfort after standby or even comfort after comfort) or after a changeover of the operating type (e.g. heating to cooling), the last changed setpoint value is discarded and replaced with the value parameterised in the ETS or specified in the mode menu in the Gira G1.

The setpoint temperature values for standby or night mode for heating or cooling are stored in volatile memory.

### Setpoint offset with absolute setpoint specification

In addition to specifying individual setpoint temperature values through the ETS or the setpoint value object, the user can shift the setpoint on the Gira G1. The setpoint can be moved either upwards or downwards by tapping on the ⊕ and ⊖ buttons. Whenever the button is tapped, the setpoint value is moved one step up or down. The offset adjustment can be made continuously by pressing and holding the button. The value of a step can be adapted to suit the application. The step width can be configured in the ETS using the "Step width of the setpoint offset" parameter in the "Room temperature controller (RTC) -> Setpoint values" parameter node (0.1 K / 0.2 K / 0.5 K / 1.0 K).

In addition to operation using the buttons ⊕ or ⊖, the detail view still offers the option of setting the setpoint value of the operating mode (comfort, standby, night) for the active operating type (heating or cooling) using an operable scale.

Value specifications can be triggered by continuously dragging the scale or tapping on the desired setpoint value on the scale. The scale increases immediately and continually up to the desired point. At the same time, the device emits the setpoint value specified in this way to the KNX. No values are transmitted to the KNX during the dragging movement. The controller rounds the setpoint temperature set to the specified step width of the setpoint offset.

The setpoint offset has an immediate effect on the "Setpoint value active operating mode" object, and therefore only has a direct effect on the specified setpoint temperature of the active operating mode. The last setpoint temperature specified via the KNX or ETS is initially overwritten by an offset. The offset setpoint value for the active operating mode only remains active temporarily. It is reset to the starting value if the operating mode or the operating type is changed or a device reset is carried out.

The setpoint temperatures of other operating modes for heating or cooling are not affected by a shift in the setpoint value for a specific operating mode. For example, if the setpoint temperature for comfort mode for heating is moved, the other setpoint values for night or standby mode for heating and cooling are not changed. If you wish to offset these as well, the temperature values must be moved individually.

The "Setpoint value active operating mode" object can be bidirectional if required (set "Transmit" flag!). This means that it is possible to use this object to report the setpoint temperature of an operating mode resulting from a setpoint offset back to the bus.

A setpoint offset does not have an impact on the setpoint temperature values for frost or heat protection.

### Sending the setpoint temperature

The setpoint temperature specified for the active operating mode can be emitted to the KNX via the 2-byte "Setpoint temperature" object. The "Transmit in case of temperature change of" parameter in the "Room temperature controller (RTC) -> Setpoint values" parameter node specifies the temperature value in kelvin by which the setpoint value must change before the setpoint temperature value is automatically emitted through the object. Temperature value changes of between 0.1 K and 3.0 K are possible. Here, a setting of "0" deactivates automatic transmission of the setpoint temperature.

The setpoint value can also be transmitted cyclically. The "Cyclical transmission of setpoint temperature" parameter defines the cycle time (1 to 240 minutes). A value of "0" deactivates cyclical transmission of the setpoint temperature value. If cyclical transmission is deactivated and automatic transmission is switched off, no telegrams regarding setpoint temperature are emitted upon change!

Placing the "Read" flag in the "Setpoint temperature" object makes it possible to read out the current setpoint value. After bus voltage is restored or after reprogramming through the ETS, the object value is initialised according to the current setpoint temperature value and actively transmitted to the bus.

## 7.6. Variable and status output

### Corrected variable objects

The format of the corrected variable objects is defined depending on the control algorithm selected for the heating and/or cooling mode. Corrected variable objects of 1 bit or 1 byte in size are created in the ETS. The control algorithm calculates the corrected variables at an interval of 30 seconds, and passes them on via the objects. In the case of pulse width-modulated PI control (PWM), the corrected variable is only updated at the end of a cycle time, if necessary.

Possible object data formats for the corrected variables, separately for both operating modes, are:

- Constant PI control: 1 byte
- Switching PI control: 1 bit + 1 byte (e.g. for status display upon visualisation)
- Switching 2-point control: 1 bit

Depending on the set operating mode, the controller can control heating and/or cooling systems, as well as determining corrected variables and issuing them via separate objects. A distinction is made between two different cases in the mixed operating mode of "Heating and cooling":

- Case 1: Heating and cooling system are two separate systems
  In this case, the checkmark beside the "Send heating and cooling variable to common object" parameter in the "Room temperature controller (RTC) -> Controller in general" parameter node should not be set. Separate objects are available for each corrected variable, through which the individual systems can be controlled separately.
  With this setting it is possible to define separate control types for heating or for cooling.
- Case 2: Heating and cooling system is a combined system
  In this case, the checkmark beside the "Send heating and cooling variable to common object" parameter can be placed if required. This means that the corrected variables for heating and cooling are sent to the same object.
  With this setting it is only possible to define the same control type for heating and for cooling as in this case the control and data format must be identical. The control parameters ("Type of heating / cooling") must still be defined separately for heating or cooling mode.
  A combined corrected variable object may be necessary if, for example, a one-pipe system (combined heating and cooling unit) is to be used for both heating and cooling. First the temperature of the medium in the one-pipe system must be changed through the system controller. Then the object is used to set the operating mode (often with a one-pipe system cold water is used for cooling in the summer and hot water is used for heating in the winter).

If necessary, the corrected variable can be inverted before it is emitted to the KNX. When output via a combined object, the corrected variable value is issued in an inverted manner according to the object data format using the parameters "Output of correcting variable heating", "Output of correcting variable cooling" or "Output of correcting variables...".

The following applies:

for constant variables:

-> not inverted: variable 0% ... 100%, value 0 ... 255

-> inverted: variable 0% ... 100%, value 255 ... 0

for switching variables:

-> not inverted: variable Off / On, value 0 / 1

-> inverted: variable Off / On, value 1 / 0

### Automatic transmission

The type of control is differentiated for automatic transmission of the corrected variable telegrams:

- Constant PI control:
  With constant PI control, the room temperature controller cyclically calculates a new corrected variable every 30 seconds and outputs it to the KNX with a 1-byte value object. In the process, the change interval for emission of new corrected variables to the bus can be specified in percent by the "Automatic transmission after change of" parameter in the "Room temperature controller (RTC) -> Variable and status output" parameter node. The change interval can be parameterised to "0" so that automatic transmission is not carried out for a corrected variable change.
  In addition to the corrected variable output in case of a change, the current corrected variable value can be cyclically output to the bus. In the process, corrected variable telegrams are output in accordance with the active value after a parameterisable cycle time in addition to the change times to be expected. This ensures that telegrams are received within the monitoring time with cyclical safety monitoring of the corrected variable in the servo or in the controlled switching actuator. The time interval specified by the "Cycle time for automatic sending" parameter should match the monitoring time in the actuator (it is preferable to parameterise the cycle time in the controller shorter). The cyclical transmission of the corrected variable is deactivated by the setting "0".
  In the case of constant PI control it should be noted that no corrected variable telegrams are emitted if cyclical transmission is deactivated and automatic transmission upon change is switched off!
- Switching PI control (PWM):
  With switching PI control (PWM), the room temperature controller also calculates a new corrected variable internally every 30 seconds. With this type of control, the corrected variable is only updated, if necessary, at the end of a PWM cycle time. The parameters "Automatic transmission after change of" and "Cycle time for automatic sending" are not effective in terms of this control algorithm. The "Cycle time for the switching variable" parameter defines the cycle time for the PWM corrected variable signal.
- 2-point control:
  With 2-point control, the room temperature and hysteresis values are evaluated cyclically every 30 seconds, meaning that the corrected variable is only modified at these times, if necessary. Because no constant corrected variables are calculated with this control algorithm, the "Automatic transmission after change of" parameter is not effective in the case of this control algorithm.
  In addition to the corrected variable output in case of a change, the current corrected variable value can be cyclically output to the KNX. In the process, corrected variable telegrams are output in accordance with the active value after a parameterisable cycle time in addition to the change times to be expected. This ensures that telegrams are received within the monitoring time with cyclical safety monitoring of the corrected variable in the servo or in the controlled switching actuator. The time interval specified by the "Cycle time for automatic sending" parameter should match the monitoring time in the actuator (it is preferable to parameterise the cycle time in the controller shorter). The cyclical transmission of the corrected variable is deactivated by the setting "0".

### Controller status

The room temperature controller can output its current status to the KNX. Various data formats are available for this purpose. The "Status controller" parameter in the "Room temperature controller (RTC) -> Variable and status output" parameter node releases the status message and defines the status format:

- "KNX compliant":
  The KNX compliant controller status feedback is harmonised independent of the manufacturer and comprises 3 communication objects. The 2-byte "KNX Status" object (DPT 22.101) shows elementary basic functions of the controller (see Table 5). This object is complemented by the two 1-byte objects "KNX status operating mode" and "KNX status forced operating mode" (DPT 20.102), which report the operating mode actually set in the controller. The two latter objects are generally used to ensure that controller extensions are able to display the controller operating mode correctly in the KNX compliant status display. Consequently, these objects must be connected with controller extensions if the KNX compliant status feedback is configured.

| Bit of status telegram | Meaning |
|---|---|
| 0 | Controller error status ("0" = no error / "1" = error) |
| 1 | Not used (permanently "0") |
| 2 | Not used (permanently "0") |
| 3 | Not used (permanently "0") |
| 4 | Not used (permanently "0") |
| 5 | Not used (permanently "0") |
| 6 | Not used (permanently "0") |
| 7 | Not used (permanently "0") |
| 8 | Operating mode ("0" = cooling / "1" = heating) |
| 9 | Not used (permanently "0") |
| 10 | Not used (permanently "0") |
| 11 | Not used (permanently "0") |
| 12 | Controller locked (dew-point mode) ("0" = controller enabled / "1" = controller locked) |
| 13 | Frost alarm ("0" = frost protection temperature exceeded / "1" = frost protection temperature dropped below) |
| 14 | Heat alarm ("0" = heat protection temperature exceeded / "1" = heat protection temperature dropped below) |
| 15 | Not used (permanently "0") |

Table 5:Bit coding of 2-byte KNX compliant status telegram

- "Controller in general":
  The general controller status combines major status information of the controller in two 1-byte communication objects. The "Controller status" object contains basic status information (see Table 6). The "Status message additional" object collects additional information bit-oriented, which is not available via the "Controller status" object (see Table 7). For example, controller extensions evaluate the additional status information in order to be able to display all of the necessary controller status information on the extension Display.

| Bit of status telegram | Meaning |
|---|---|
| 0 | At "1": Comfort mode active |
| 1 | At "1": Standby mode active |
| 2 | At "1": Night mode active |
| 3 | At "1": Frost/heat protection active |
| 4 | At "1": Controller locked out |
| 5 | At "1": Heating, at "0": Cooling |
| 6 | At "1": Controller inactive (dead zone) |
| 7 | At "1": Frost alarm ($T_{room} \leq +5$ °C) |

Table 6:Bit coding of 1-byte status telegram

| Bit of status telegram | Meaning for "1" | Meaning for "0" |
|---|---|---|
| 0 | Normal operating mode | Forced operating mode |
| 1 | Comfort extension active | No comfort extension |
| 2 | Presence (presence detector) | No presence (presence detector) |
| 3 | Presence (presence button) | No presence (presence button) |
| 4 | Window opened | No window opened |
| 5 | Additional stage active | Additional stage not active |
| 6 | Heat protection active | Heat protection not active |
| 7 | Controller locked out (dew-point mode) | Controller not locked out |

Table 7:Bit coding of 1-byte additional status telegram

After a reset, the status objects are only updated following the initialisation phase. After that, they are updated cyclically every 30 seconds, in line with the controller's corrected variable calculation. Telegrams are then only sent to the KNX if the status changes.

**Variable limiting**

Variable limiting can be configured in the ETS if desired. Variable limiting allows you to restrict calculated corrected variables of the controller to the range limits of "Minimum" and "Maximum". The limits are set in the ETS and, if variable limiting is active, cannot be exceeded or fallen short of during operation of the device. Various limits can be specified for heating and cooling.

Variable limiting has no effect in the case of "2-point control" and "Send heating and cooling variables through common object"! Variable limiting can be configured in the ETS, but has no function.

The "Variable limiting" parameter on the "Room temperature controller (RTC) -> Variable and status output" parameter page defines how the limiting function works. Variable limiting can either be activated and deactivated using the 1-bit "Variable limiting" communication object or be permanently active. When controlling using the object it is possible to have variable limiting automatically activated by the controller when bus voltage is restored or after an ETS programming procedure. The "Variable limiting after reset" parameter defines the initialisation behaviour. If the setting is "disabled", variable limiting is not automatically activated after a device reset. A "1" telegram must be received via the "Variable limiting" object for the limiting to be activated. If the setting is "activated", the controller automatically activates variable limiting after a device reset. A "0" telegram must be received via the "Variable limiting" object for the limiting to be deactivated. Limiting can be switched on or off through the object at any time.

In the case of permanent variable limiting, the initialisation behaviour after a device reset cannot be configured separately as limitation is always active. In this case, it is also not possible to configure an object.

As soon as variable limiting is active, calculated corrected variables are restricted in accordance with the limits from the ETS. The behaviour in terms of the minimum or maximum corrected variable can be described as follows:

- Minimum correcting variable:
  The "Minimum correcting variable" parameter specifies the lower variable limit. The setting can be implemented in 5% increments within a range of 5% ... 50%. If variable limiting is active, it is not possible to fall below the set minimum variable value. Should the controller calculate lower corrected variables, it sets the configured minimum variable. The controller emits 0% corrected variables if heating or cooling energy no longer needs to be requested.
- Maximum correcting variable:
  The "Maximum correcting variable" parameter specifies the upper variable limit. The setting can be implemented in 5% increments within a range of 55% ... 100%. If variable limiting is active, it is not possible to exceed the set maximum variable value. Should the controller calculate higher corrected variables, it sets the configured maximum variable.

If the limiting is removed, the controller only updates the last calculated corrected variable to the unrestricted values automatically once the next calculation interval for the variables (30 seconds) has passed.

If the controller is performing valve protection, the variable limiting is temporarily deactivated in order to fully utilise the valve's travel path.

Having variable limiting activated has a negative impact on the control result, particularly if the variable range is highly restricted. Control deviation is to be expected.

**Special case: 100% variable (clipping mode)**

If, in the case of PI control, the calculated corrected variable of the controller exceeds the physical limits of an actuator, i.e. the calculated corrected variable is greater than 100%, the variable is set to the maximum value (100%), therefore limiting it. This special and necessary form of behaviour is also known as "clipping". In the case of PI control, the corrected variable may reach a value of "100%" if the deviation between the room temperature and the setpoint temperature is great or the controller needs a long time to adjust the input heating or cooling energy to the setpoint value. The controller may evaluate this state in a specific way and react differently to it.

The "Behaviour for correcting variable = 100% (clipping mode)" parameter on the "Room temperature controller (RTC) -> Variable and status output" parameter page is only visible if PI control is used for heating and/or cooling. The "Behaviour for correcting variable = 100% (clipping mode)" defines how the PI controller works when the variable is 100%:

- Setting "Hold 100% until setpoint = actual value, then 0%":
  The controller holds the maximum variable, without interruption, until the room temperature (actual value) reaches the setpoint temperature. After that, it suddenly reduces the variable to 0% (controller reset).
  One thing that is advantageous with this control behaviour is that it achieves sustainable heating in very cool rooms or effective cooling in overheated environments, due to overshooting of the setpoint value. One disadvantage is that these major swings in room temperature can be perceived as problematic.
- Setting "Hold 100% as required, then reduce":
  The controller only holds the maximum variable for as long as is necessary. It then reduces the variable in accordance with the PI algorithm. The benefit of this control property is that the room temperature does not, or does not significantly, exceed the setpoint temperature. What is disadvantageous is the fact that this control principle increases the tendency to fluctuate around the setpoint value.

Which of the aforementioned functional principles is used often depends on what type of heating or cooling system is being used (underfloor heating, radiators, fan coil units, cooling ceilings, etc.) and how effective these systems are. It is recommended that you select the setting "Hold 100% until setpoint = actual value, then 0%" (standard setting). Only if this control behaviour has a disadvantageous effect on how the people in a room feel should the setting "Hold 100% as required, then reduce" be used.

Clipping may also occur if variable limiting is active (maximum variable). In this case, if the variable reaches 100% internally, the controller only sends the maximum corrected variable as per the ETS configuration to the KNX. The clipping (switch-off upon setpoint = actual or reduce) does, however, take place.

The clipping mode has no effect with "2-point control"!
The "Behaviour for correcting variable = 100%" parameter can be configured in the ETS, but has no function.

## 7.7. Blocking functions of the room temperature controller

### Block controller

In certain operating modes it may be necessary to deactivate the room temperature control. For example, the control can be switched off in the dew-point mode of a cooling system or during maintenance work on the heating or cooling system. The "Switch off controller (dew-point mode)" parameter in the "Room temperature controller (RTC) -> Controller functionality" parameter node enables the 1-bit "Block control-

ler" object with the setting "via bus". In addition, the controller blocking function can be switched off with the "no" setting.

If a "1" telegram is received via the blocker, the room temperature control is completely deactivated. In this case, the corrected variables of the control circuit are equal to "0" (wait for 30 s updating interval of the corrected variables) and the "Dew-point mode"? icon appears on the device Display. Operation of the controller is possible in this case, however.

A block is always deleted after a reset (restoration of bus voltage, ETS programming procedure)!

### Block controller operation

You have the option of blocking operation of the room temperature controller. If operation is blocked, the controller can no longer be used. Operation via the KNX, e.g. through the operating mode or setpoint value objects, is, however, still possible. Active blocking of operation is indicated in the Display by the icon.

The "Controller operation lockable" parameter in the "Room temperature control -> Controller functionality" branch of parameters can be used to set whether operation is always impossible (setting: "always locked") or can be introduced through the "Block controller operation" object (setting: "via bus"). If the setting is "via bus", operation is deactivated upon receipt of a "1" telegram by the object. Consequently, a "0" telegram releases operation again. Controller operation has no reaction if blocking is in place.

Activated blocking of controller operation does not affect the room temperature control itself, i.e. the control algorithm continues to work and generates corrected variables and status messages.

A block on controller operation is always deleted after a reset.

## 7.8. Valve protection

In order to prevent lime scale accumulation or jamming of the radiator or cooling system control valves used, cyclical valve protection can be carried out. The "Valve protection" parameter in the "Room temperature controller (RTC) -> Controller functionality" parameter node activates the valve protection.

This protective function is generally not launched for inactive corrected variable objects, i.e. objects that have not requested any heating or cooling energy in the last 24 hours. For these objects, the controller cyclically sets the variable to the maximum value once a day for a duration of 5 minutes, taking into account the following parameterisation:

Variable output not inverted:

-> 1-bit variable: "1", 1-byte variable: "255" Variable output inverted:

-> 1-bit variable: "0", 1-byte variable: „0“

This means that even valves that have been closed for a long time are briefly opened on a regular basis.

Controller lock-out does not affect valve protection. This means that valve protection is carried out even if the controller is blocked.

The controller checks the 24 h time cycle for valve protection using its internal clock. Valve protection is carried out each day at 8am, using a time-synchronised clock. If the time signal has not been synchronised via the bus for an extended period, the clock continues with the expected rate. It can happen that the point of valve protection continually shifts if the clock is not synchronised.

# 8. Parameters

## 8.1. General

| Description | Values | Comment |
|---|---|---|
| Display time in status bar of the Gira G1? | **active**<br>inactive | Displays the date and time in the status bar of the Gira G1, if active. |
| Display outdoor temperature in status bar of the Gira G1? | active<br>**inactive** | This parameter defines whether the outdoor temperature is displayed in the status bar. The outdoor temperature is displayed in the Gira G1 status bar with the unit "°C". |
| Use room temperature measurement? | active<br>**inactive** | Activates or deactivates room temperature measurement. |
| Use room temperature controller? | active<br>**inactive** | The controller function block integrated in the device can be switched on or off. The setting of this parameter considerably affects the functioning and other parameters and objects displayed in the ETS.<br>**"inactive":** The controller function block is fully switched off. Room temperature control cannot be executed by the device.<br>**"active":** The controller function block is switched on. The room temperature controller area is visible and contains other pages with settings for the room temperature controller. |
| Time mode | **Not used** | In this case, the Gira G1 queries an NTP server at 0.europe.pool.ntp.org at intervals of 10 min, therefore synchronising the system time. |
| | Gira G1 is time server | If "Gira G1 is time server" is selected, the date and time can be queried and cyclically transmitted to the bus.<br>The time and date are transmitted each full minute and after synchronisation |
| | Gira G1 is time client | If "Gira G1 is time client" is selected, the date and time of the device can be set via the bus.<br>Only one time server may be assigned to the Gira G1. |
| Display restart? | active<br>**inactive** | If this parameter is activated, the device can be restarted using a communication object. |
| Display KNX statuses? | active<br>**inactive** | If this parameter is activated, the communication objects are enabled for programming mode and bus voltage.<br>This allows the bus status and the programming button to be queried. The programming mode can also be set using a communication object. |
| Use Gira door communication? | active<br>**inactive** | Enables the support for the Gira DCS IP gateway, therefore activating the display of the door communication function in the Gira G1. |

| | | |
|---|---|---|
| Door communication type | Gira door communication system | Switches the support for the Gira TKS-IP gateway off and thus activates the display of the door communication function on the Gira G1. |
| | SIP door communication system | Switches the support for a SIP door communication system and thus activates the display of the door communication function on the Gira G1. |
| | **None** | The door communication function should not be used on the Gira G1. |
| Use floor button in KNX? | active<br>**inactive** | Enables a communication object through which a floor-call can be triggered. |
| Name of the floor button in user interface | **Floor button** | Defines the name of the floor button. This name is displayed in the Gira G1 when a floor-call comes in. |
| Melody assigned to floor-call | **1**...10 | Selects one of 10 melodies for the floor-call from a list. |
| Use Gira Weather Service? | active<br>**inactive** | Activates or deactivates display of the Gira Weather Service. |
| Reset all user data during an ETS programming procedure? | active<br>**inactive** | All user settings configured via the app, e.g. timers, are deleted.<br>Important: This option deletes all changes made by the user in the Home view, as well as all timers. |

### 8.2. "Room temperature measurement" / "General"

| Description | Values | Comment |
|---|---|---|
| Display room temperature in status bar of the Gira G1? | active<br>**inactive** | This parameter defines whether the room temperature is displayed in the status bar. |
| Use unit text? | **active** | If the "Display room temperature in status bar of the Gira G1?" parameter is active, the unit is always displayed. The unit text for the temperature is set to "°C" and cannot be changed. |
| Text for unit | **°C** | The unit text for the temperature is set to "°C" and cannot be changed. |
| Sensor selection | **Internal sensor only** | The actual temperature is detected exclusively via the temperature sensor module.<br>Temperature detection via the temperature sensor module is only supported in the Gira G1 from index status I09. |
| | Received temperature value only | The actual temperature is detected exclusively via an external temperature sensor. The measured temperature value can be read out via the KNX using the 2-byte "Received temperature" object as required. Temperature values are shown if a valid value has been received or queried. |

| | | |
|---|---|---|
| | internal sensor and received temperature value | The actual temperature is detected based on the relationship specified in "Measured value generation to be received internally". |
| Measured value generation to be received internally | 10% to 90%...**50% to 50%**...90% to 10% | Only visible if the temperature value (complete or proportional) is received via the bus. This parameter defines the relationship between the internal temperature value and the received temperature value. |
| Comparison of received temperature value | -10.0...**0.0**...10.0K | Only visible if the temperature value (complete or proportional) is received via the bus. Determines the value with which the received room temperature measured by the external sensor is compared. |
| Query time for received temperature value | **0...**240min<br>(0 = automatic query deactivated) | Only visible if the temperature value (complete or proportional) is received via the bus. This is where the query period for the temperature value to be received by the KNX is stipulated. If the setting is "0", the temperature value is not automatically queried by the device. In this case, the communication partner (e.g. controller) must independently send out its temperature value. |
| Automatic transmission of actual temperature after change of | **0.0**...3.0K<br>(0 = automatic transmission deactivated) | Determines the size of change in value of room temperature after which the current values are automatically emitted to the KNX via the "Actual temperature" object. If the setting is "0", there is no automatic transmission. |
| Cyclical transmission of actual temperature | **0...**240min<br>(0 = cyclical transmission deactivated) | This parameter defines whether and at what time the determined room temperature should be cyclically emitted via the "Actual temperature" object. If the setting is "0", cycling transmission is deactivated |

### 8.3. "Room temperature controller (RTC)" / "General"

| Description | Values | Comment |
|---|---|---|
| User interface description | **Temperature** | Name that is visible in the user interface. |
| Select function icon | **014 - Temperature** | This parameter defines the function icon for the room temperature controller. The icon is displayed in the user interface of the Gira G1.<br>A preview of the selected icon is displayed below this parameter.<br>List of available icons, see Page 154. |
| Use heating timers | **active**<br>inactive | This parameter enables the "Heating timer" button on the Gira G1 user interface. |
| Overwrite user data during an ETS programming procedure? | active<br>**inactive** | All settings of the internal temperature controller are deleted. |
| Operating mode | **Heating** | Corrected variables of a heating system are used. |

| | | |
|---|---|---|
| | Cooling | Corrected variables of a cooling system are used. |
| | Heating and cooling | Mixed mode. A communication object facilitates toggling between the two operating modes. |
| Send heating and cooling variable to common object? | active<br>**inactive** | If the parameter is set to "Active", the heating or cooling variable is sent to a common object. This function is used when the same heating system is used to cool a room in summer and heat it in winter. This parameter is only visible in the mixed operating mode "Heating and cooling". |
| Type of heating control | **Constant PI control**<br>switching PI control (PWM)<br>switching 2-point control | Selection of a control algorithm (PI or 2-point) with data format (1 byte or 1 bit) for the heating system. |
| Type of heating | **Hot water heating (5 K / 150 min)**<br>Underfloor heating (5 K / 240 min)<br>Electric heating (4 K / 100 min)<br>Fan coil unit (4 K / 90 min)<br>Split unit (4 K / 90 min)<br>via control parameters | Adjustment of the PI algorithm to different heating systems with predefined values for the control parameters of "Proportional area" and "Reset time". The "via control parameters" setting makes it possible to set the control parameters within certain limits, in deviation from the predefined values. This parameter is only visible if "Type of heating control = PI control". |
| Proportional area heating | 1.0...**5.0**...10.0 K | Separate setting of the "Proportional area" control parameter. This parameter is only visible if "Type of heating = via control parameters" and the heating control type is "PI control". |
| Reset time heating | 0...**240** min<br>(0 = no readjustment time) | Separate setting of the "Reset time" control parameter. This parameter is only visible if "Type of heating = via control parameters" and the heating control type is "PI control". |
| Lower hysteresis of the 2-point controller heating | -10.0...**-0.5 K** | Definition of the lower hysteresis (switch-on temperatures) for heating. This parameter is only visible if "Type of heating control = switching 2-point control". |
| Upper hysteresis of the 2-point controller heating | **0.5**...10.0 K | Definition of the upper hysteresis (switch-off temperatures) for heating. This parameter is only visible if "Type of heating control = switching 2-point control". |

| | | |
|---|---|---|
| Type of cooling control | **Constant PI control**<br>switching PI control (PWM)<br>switching 2-point control | Selection of a control algorithm (PI or 2-point) with data format (1 byte or 1 bit) for the heating system. |
| Type of cooling | **Cooling ceiling (5 K / 240 min)**<br>Fan coil unit (4 K / 90 min)<br>Split unit (4 K / 90 min)<br>via control parameters | Adjustment of the PI algorithm to different cooling systems with predefined values for the control parameters of "Proportional area" and "Reset time". The "via control parameters" setting makes it possible to set the control parameters within certain limits, in deviation from the predefined values. This parameter is only visible if "Type of heating control = PI control". |
| Proportional area cooling | 1.0...**5.0**...10.0 K | Separate setting of the "Proportional area" control parameter. This parameter is only visible if "Type of cooling = via control parameters" and the cooling control type is "PI control". |
| Reset time cooling | 0...**240** min<br>(0 = no readjustment time) | Separate setting of the "Reset time" control parameter. This parameter is only visible if "Type of cooling = via control parameters" and the cooling control type is "PI control". |
| Lower hysteresis of the 2-point controller cooling | -10.0...**-0.5 K** | Definition of the lower hysteresis (switch-off temperatures) for basic cooling. This parameter is only visible if "Type of cooling control = switching 2-point control". |
| Upper hysteresis of the 2-point controller cooling | **0.5**...10.0 K | Definition of the upper hysteresis (switch-on temperatures) for cooling. This parameter is only visible if "Type of cooling control = switching 2-point control". |
| Operating mode after reset | **Restore operating mode after reset**<br>Comfort mode<br>Standby mode<br>Night mode<br>Frost/heat protection mode | This parameter defines which operating mode is set directly after a device reset. If "Restore operating mode before reset": The operating mode set before a reset (normal priority) is set again after the device's initialisation phase. Operating modes set before the reset using a function with a higher priority (forced, window status, presence status) are not updated. |
| Operating mode heating/cooling after reset | **Heating**<br>Cooling<br>Operating mode before reset | This is where the preset operating mode after restoration of bus voltage is defined. |
| Delay window status | **0**...240 min<br>(0 = no delay) | This parameter defines the delay time for the window status. |

### 8.4. "Room temperature controller (RTC)" / "Setpoint values" page

| Description | Values | Comment |
|---|---|---|
| Comfort mode setpoint temperature (heating) | 7.0...**21.0**...40.0 °C | The setpoint temperatures for the comfort, standby and night mode are independent of each other. Various temperature values in the range from +7.0 °C to +40.0 °C can be specified for each operating mode and operating type in the ETS. The ETS does not validate the temperature value. For example, it is possible to specify lower setpoint temperatures for the cooling mode than for the heating mode or lower temperatures for the comfort mode than for the standby mode. After start-up through the ETS, the setpoint temperatures can be changed via the KNX by temperature telegrams. The "Setpoint value active operating mode" communication object is available for this purpose. Specification of setpoint temperature for comfort mode heating. |
| Standby mode setpoint temperature (heating) | 7.0...**19.0**...40.0 °C | Specification of setpoint temperature for standby mode heating. |
| Night mode setpoint temperature (heating) | 7.0...**17.0**...40.0 °C | Specification of setpoint temperature for night mode heating. |
| Comfort mode setpoint temperature (cooling) | 7.0...**23.0**...40.0 °C | Specification of setpoint temperature for comfort mode cooling. |
| Standby mode setpoint temperature (cooling) | 7.0...**25.0**...40.0 °C | Specification of setpoint temperature for standby mode cooling. |
| Night mode setpoint temperature (cooling) | 7.0...**27.0**...40.0 °C | Specification of setpoint temperature for night mode cooling. |
| Allow changes to standard setpoints for each operating mode in user interface? | **active**<br>inactive | This setting permits modification of setpoint values via the user interface. |
| Minimum setpoint temperature | 7.0...**17.0**...40.0 °C | Defines the minimum setpoint temperature. This should be lower than or equal to the lowest other set setpoint value. |
| Maximum setpoint temperature | 7.0...**27.0**...40.0 °C | Defines the maximum setpoint temperature. This should be higher than or equal to the highest other set setpoint value. |
| | Please observe that the minimum and maximum setpoint temperature is always configured smaller than or equal to, or greater than or equal to all other setpoint temperatures. | |
| Step width of the setpoint offset | **0,1**..0,2..0,5..1.0 K | This parameter defines the value of a step in the basic setpoint value offset. The controller extension must work with the same step width as the controller main unit. |

| | | |
|---|---|---|
| Anti-freeze setpoint temperature | **7.0**...40.0 °C | This parameter defines the setpoint temperature for frost protection. The parameter is only visible in the operating mode "Heating" or "Heating and cooling". |
| Heat protection setpoint temperature | 7.0...**35.0**...40.0 °C | This parameter defines the setpoint temperature for heat protection. The parameter is only visible in the operating mode "Cooling" or "Heating and cooling". |
| Transmit in case of temperature change of | 0.0...**0.1**...3.0 K<br>(0.0 = automatic transmission deactivated) | Determines the size of change in value of setpoint value after which the current value is automatically transmitted to the bus via the "Setpoint temperature" object. If the setting is "0", the setpoint temperature is not emitted automatically in the event of a change. |
| Cyclical transmission of setpoint temperature | **0**...240 min<br>(0 = cyclical transmission deactivated) | This parameter defines whether the setpoint temperature is to be cyclically emitted via the "Setpoint temperature" object. Definition of cycle time by means of this parameter. If the setting is "0", the setpoint temperature is not emitted cyclically. |

### 8.5. "Room temperature controller (RTC)" / "Variable and status output" page

| Description | Values | Comment |
|---|---|---|
| Automatic transmission after change of | 0...**3...**10%<br>(0 = automatic transmission deactivated) | This parameter determines the size of change in corrected variable value after which constant corrected variable telegrams are automatically transmitted via the corrected variable objects. This parameter only affects corrected variables parameterised as "Constant PI control" and 1-byte additional corrected variable objects of "switching PI control (PWM)". If the setting is "0", corrected variable telegrams are not emitted depending on corrected variable change. |
| Cycle time for the switching variable | 1...**15...**240 min | This parameter defines the cycle time for pulse width-modulated variables (PWM). This parameter only affects corrected variables that are parameterised through "Switching controls". |
| Cycle time for automatic sending | 0...**10**...240 min<br>(0 = cyclical transmission deactivated) | This parameter defines the time interval for cyclical transmission of corrected variables across all corrected variable objects. If the setting is "0", corrected variable telegrams are not emitted cyclically. |
| Output of correcting variable heating | **normal (energised means open)**<br>inverted (energised means closed) | Here is where you specify whether the corrected variable telegram for heating is to be output normally or inverted. This parameter is only visible if the operating mode "Heating" or "Heating and cooling" is configured. |
| Output of correcting variable cooling | **normal (energised means open)**<br>inverted (energised means closed) | Here is where you specify whether the corrected variable telegram for cooling is to be output normally or inverted. This parameter is only visible if the operating mode "Cooling" or "Heating and cooling" is configured. |

| | | |
|---|---|---|
| Variable limiting | **disabled**<br>permanently activated<br>activated via object | Variable limiting allows you to restrict calculated corrected variables of the controller to the range limits of "Minimum" and "Maximum". The limits are set in the ETS and, if variable limiting is active, cannot be exceeded or fallen short of during operation of the device. The "Variable limiting" parameter defines how the limiting function works. Variable limiting can either be activated and deactivated using the 1-bit "Variable limiting" communication object or be permanently active. |
| Variable limiting after reset | **disabled**<br>activated | When controlling using the object it is possible to have variable limiting automatically activated by the controller when bus voltage is restored or after an ETS programming procedure. This parameter defines the initialisation behaviour.<br>If the setting is "disabled", variable limiting is not automatically activated after a device reset. A "1" telegram must be received via the "Variable limiting" object for the limiting to be activated.<br>If the setting is "activated", the controller automatically activates variable limiting after a device reset. A "0" telegram must be received via the "Variable limiting" object for the limiting to be deactivated. Limiting can be switched on or off through the object at any time. This parameter is only visible if "Variable limiting = can be activated via object"! |
| Minimum correcting variable heating | **5**...50 | The "Minimum correcting variable" parameter specifies the lower variable limit for heating. If variable limiting is active, it is not possible to fall below the set minimum variable value. Should the controller calculate lower corrected variables, it sets<br>the configured minimum variable. The controller emits 0% corrected variables if heating energy no longer needs to be requested. |
| Minimum correcting variable heating / cooling | **5**...50 | The "Minimum correcting variable" parameter specifies the lower variable limit for heating and cooling. If variable limiting is active, it is not possible to fall below the set minimum variable value. Should the controller calculate lower corrected variables, it sets the configured minimum variable. The controller emits 0% corrected variables if heating or cooling energy no longer needs to be requested. This parameter is only visible if the operating mode "Heating and cooling" is configured and "Send heating and cooling variable to common object?" is active. |
| Maximum correcting variable heating | 55...**95**...100 | The "Maximum correcting variable" parameter specifies the upper variable limit for heating. If variable limiting is active, it is not possible to exceed the set maximum variable value. Should the controller calculate higher corrected variables, it sets the configured maximum variable. |

| | | |
|---|---|---|
| Maximum correcting variable heating / cooling | 55...**95...**100 | The "Maximum correcting variable" parameter specifies the upper variable limit for heating and cooling. If variable limiting is active, it is not possible to exceed the set maximum variable value. Should the controller calculate higher corrected variables, it sets the configured maximum variable. This parameter is only visible if the operating mode "Heating and cooling" is configured and "Send heating and cooling variable to common object?" is active. |
| Minimum correcting variable cooling | **5**...50 | The "Minimum correcting variable" parameter specifies the lower variable limit for cooling. If variable limiting is active, it is not possible to fall below the set minimum variable value. Should the controller calculate lower corrected variables, it sets the configured minimum variable. The controller emits 0% corrected variables if cooling energy no longer needs to be requested. |
| Maximum correcting variable cooling | 55...**95...**100 | The "Maximum correcting variable" parameter specifies the upper variable limit for cooling. If variable limiting is active, it is not possible to exceed the set maximum variable value. Should the controller calculate higher corrected variables, it sets the configured maximum variable. |
| Heating message | active<br>**inactive** | Depending on the set operating mode, a separate object can be used to signal whether the controller is currently requesting heating energy for the control circuit, and therefore whether heating is actively taking place. The "Active" setting here enables the message function for heating. |
| Cooling message | active<br>**inactive** | Depending on the set operating mode, a separate object can be used to signal whether the controller is currently requesting cooling energy for the control circuit, and therefore whether cooling is actively taking place. The "Active" setting here enables the message function for cooling. |
| Status controller | **No status**<br>KNX compliant<br>Controller in general | The room temperature controller is capable of outputting its current status to the KNX. Various data formats are available for this purpose. This parameter releases the status message and defines the status format. |

| | | |
|---|---|---|
| Behaviour for correcting variable = 100% (clipping mode PI control) | **Hold 100% until setpoint = actual value, then 0%**<br>Hold 100% as required, then reduce | If, in the case of PI control, the calculated corrected variable of the controller exceeds the physical limits of an actuator, i.e. the calculated corrected variable is greater than 100%, the variable is set to the maximum value (100%), therefore limiting it. In the case of PI control, the corrected variable may reach a value of "100%" if the deviation between the room temperature and the setpoint temperature is great or the controller needs a long time to adjust the input heating or cooling energy to the setpoint value. The controller may evaluate this state in a specific way and react differently to it. This parameter defines the functioning of the PI controller when the variable is 100%.<br>Setting "Hold 100% until setpoint = actual value, then 0%": The controller holds the maximum variable, without interruption, until the room temperature (actual value) reaches the setpoint temperature. After that, it suddenly reduces the variable to 0% (controller reset). One thing that is advantageous with this control behaviour is that it achieves sustainable heating in very cool rooms or effective cooling in overheated environments, due to overshooting of the setpoint value. One disadvantage is that these major swings in room temperature can be perceived as problematic.<br>Setting "Hold 100% as required, then reduce": The controller only holds the maximum variable for as long as is necessary. It then reduces the variable in accordance with the PI algorithm. The benefit of this control property is that the room temperature does not, or does not significantly, exceed the setpoint temperature. What is disadvantageous is the fact that this control principle increases the tendency to fluctuate around the setpoint value. |

### 8.6. "Room temperature controller (RTC)" / "Controller functionality" page

| Description | Values | Comment |
|---|---|---|
| Presence detection | **none**<br>Presence button<br>Presence detector | If the setting is "none", presence mode is deactivated.<br>If the setting is "Presence button", presence detection is carried out through the "Presence" button on the device or through the presence object (e.g. other touch sensors). If the "Presence" button is pressed in night mode or frost/heat protection, the comfort extension is activated. If the "Presence" button is pressed in standby mode, the controller activates comfort mode for the duration of presence mode.<br>If the setting is "Presence detector", presence detection takes place through an external presence detector connected to the presence object. If presence is detected, comfort mode is called up. Comfort mode remains active until the presence detector no longer detects any movement. |
| Duration of comfort extension | 0...**30**...240 min | If the "Presence" button is pressed in night mode or frost/heat protection, the controller switches to comfort mode for the period of time specified here. It switches back off again automatically at the end of this time. If the setting is "0", the comfort extension is switched off, meaning that it cannot be activated from night mode or frost/heat protection mode. In this case, the operating mode is not changed, although the presence function is activated. This parameter is only visible if the presence detection is configured to "Presence button". |
| Controller operation lockable | **No**<br>always locked<br>via bus | You have the option of blocking operation of the room temperature controller. If operation is blocked, the controller can no longer be used via the buttons on the device. Operation via the bus, e.g. through the operating mode or setpoint value objects, is, however, still possible. This parameter allows you to define whether operation can be introduced through the "Block controller operation" object (setting: "via bus"). If the setting is "via bus", operation is deactivated upon receipt of a "1" telegram by the object. Consequently, a "0" telegram releases operation again. |
| Switch off controller (dew-point mode) | active<br>**inactive** | This parameter releases the "Block controller" object. If a controller is blocked, no further control will take place until it is released (corrected variables = 0). Active blocking of the controller (dew-point mode) is shown in the Display. |

| | | |
|---|---|---|
| Valve protection | active<br>**inactive** | In order to prevent lime scale accumulation or jamming of the radiator or cooling system control valves used, cyclical valve protection can be carried out. This parameter activates the valve protection when set to "Active". This protective function is generally not launched for inactive corrected variable outputs, i.e. outputs that have not requested any heating or cooling energy in the last 24 hours. For these outputs, the controller cyclically sets the variable to the maximum value once a day for a duration of 5 minutes. |

### 8.7. "Touch sensor" / "Overview of function folders" page

| Description | Values | Comment |
|---|---|---|
| Number of function folders | **1**...6 | Sets how many function folders can be used. Up to six function folders can be created. Each function folder can be assigned a maximum of 25 channel functions following configuration.<br>A function folder serves to structure the individual functions. In a function folder, channel functions such as switching, dimming, etc. can be organised based on various criteria and with reference to the KNX system. For example, a function folder may be a room, a floor or a subsection. Each function folder may be given a description to facilitate location at a later date. A new parameter page, on which the function folder is configured, appears for each function folder created. |

### 8.8. "Touch sensor" / "Overview of function folders" / "Function folder n" page

| Description | Values | Comment |
|---|---|---|
| Number of channels | **1** -25 | A maximum of 25 channels can be enabled in each of the up to 6 function folders. Each of the created channels can be assigned a function following configuration. For example, switching, dimming or value transmitter functions can be implemented. Each channel may be given a description to facilitate location at a later date. A new parameter page, on which the channel function is configured, appears for each channel created.<br>The channels are consecutively numbered across the function folders:<br>Function folder 1: Channels 1 to 25<br>Function folder 2: Channels 26 to 50 ...<br>Function folder 6: Channels 126 to 150 |
| Display on Home? | **active**<br>inactive | Sets whether this function folder is shown on the Home screen of the Gira G1. |
| Description of left structure node | < empty > | This text parameter describes the function folder in the ETS. |

| | | |
|---|---|---|
| User interface description | **Page** | Changes the name on the Gira G1 user interface. The name is displayed as the name of the function folder in the tile or detail view. |
| Icon selection | **121 - Function folder** | This parameter defines the icon for the function folder. The icon is displayed in the user interface of the Gira G1.<br>A preview of the selected icon is displayed below this parameter.<br>List of available icons, see Page 154. |

### 8.9. "Touch sensor" / "Overview of function folders" / "Function folder n" / "Channel n" page

| Description | Values | Comment |
|---|---|---|
| Function | no function | This is where the basic function of the channel is defined. Depending on this setting, the ETS displays different communication objects and parameters for this channel.<br>If "no function" is set, no other parameters and objects are made available for this channel, apart from the parameter "Description of left structure node". |
| Description of left structure node | <empty> | This text parameter describes the channel in the ETS. |
| User interface description | **Channel** | This text parameter describes the channel function for the user interface. The name is displayed as the name of the function in the tile or detail view. |
| Icon selection | **001 - Light** | This parameter defines the function icon for the channel. The icon is displayed in the user interface of the Gira G1.<br>A preview of the selected icon is displayed below this parameter.<br>List of available icons, see Page 154. |
| Display on Home? | active<br>**inactive** | This parameter sets whether the function is displayed on the Home screen or not |

### 8.10. Parameters of the "Switching" function

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Type of operating element | Button function<br>**Rocker function** | Buttons on the user interface can be displayed and evaluated as buttons (single-surface principle) or as rockers (double-surface principle). |
| Command when button is pressed | ON<br>OFF<br>**TOGGLE** | A "1" is always sent in the case of "ON", a "0" is always sent in the case of "OFF". In the case of "TOGGLE", the opposite value to the status is sent. If the status is "OFF", "ON" is sent, and vice versa. |

| | | |
|---|---|---|
| Select text for 1 - Switch on | **Switch on** | This parameter defines the text to be shown for switch-on in the user interface |
| Select text for 0 - Switch off | **Switch off** | This parameter defines the text to be shown for switch-off in the user interface |
| Display status? | **active**<br>inactive | This parameter activates additional parameters that define how status is handled. |
| Colour of the status displays for I - ON | Red<br>Green<br>Blue<br>**Orange**<br>Grey (OFF) | This parameter defines the colour of the ring in the detail view or the switch in the tile view (button function only) if the value is 1 (ON). |
| Colour of the status displays for 0 - OFF | Red<br>Green<br>Blue<br>Orange<br>**Grey (OFF)** | This parameter defines the colour of the ring in the detail view or the switch in the tile view (button function only) if the value is 0 (OFF). |
| Select status text for I - ON | **ON** | This parameter defines the text in the detail view or in the tile view if the value is 1 (ON). |
| Select status text for 0 - OFF | **OFF** | This parameter defines the text in the detail view or in the tile view if the value is 0 (OFF). |

### 8.11. Parameters of the "Trigger ON/OFF" function

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Command when button is pressed | **ON**<br>OFF | This parameter defines whether a 1 (ON) or a 0 (OFF) is transmitted. |
| Colour | Red<br>Green<br>Blue<br>**Orange**<br>Grey (OFF) | This parameter defines the colour upon triggering. |
| Text | **<empty>** | This parameter defines the text on the button. |

### 8.12. Parameters of the "Press / release ON/OFF" function

| Description | Values | Comment |
|---|---|---|
| Command when button is pressed | **ON**<br>OFF | This parameter defines whether a 1 (ON) or a 0 (OFF) is transmitted when the button is pressed. |

| | | |
|---|---|---|
| Colour | Red<br>Green<br>Blue<br>**Orange**<br>Grey (OFF) | This parameter defines the colour upon triggering. |
| Text | **<empty>** | This parameter defines the text on the button. |
| Command when button is released | ON<br>**OFF** | This parameter defines whether a 1 (ON) or a 0 (OFF) is transmitted when the button is released. |

### 8.13. Parameters of the "Dimming brightness value" function

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Step width | 1...**10**...50% | This parameter defines the percentage by which a step dims. |
| Time between switching and dimming command | 0.1...**0.4**...3.0 s | This parameter defines when switching takes place and when dimming takes place. If the period between pressing and releasing is less than the set value, switching takes place. If it is greater, dimming takes place. |
| Select status text for 1 - ON | **ON** | This parameter defines the text in the detail view or in the tile view if the value is 1 (ON). |
| Select status text for 0 - OFF | **OFF** | This parameter defines the text in the detail view or in the tile view if the value is 0 (OFF). |

### 8.14. Parameters of the "Dimming relative" function

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Time between switching and dimming command | 0.1...**0.4**...3.0 s | This parameter defines when switching takes place and when dimming takes place. If the period between pressing and releasing is less than the set value, switching takes place. If it is greater, dimming takes place. |
| Display status? | **active**<br>inactive | This parameter activates additional parameters that define how status is handled. |
| Select status text for 1 - ON | **ON** | This parameter defines the text in the detail view or in the tile view if the value is 1 (ON). |
| Select status text for 0 - OFF | **OFF** | This parameter defines the text in the detail view or in the tile view if the value is 0 (OFF). |

### 8.15. Parameters of the "Dimming RGB" and "Dimming RGBW" functions

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Step width | 1...**10**...50% | This parameter defines the percentage by which a step dims. |
| Time between switching and dimming command | 0.1...**0.4**...3.0 s | This parameter defines when switching takes place and when dimming takes place. If the period between pressing and releasing is less than the set value, switching takes place. If it is greater, dimming takes place. |
| Select status text for 1 - ON | **ON** | This parameter defines the text in the detail view or in the tile view if the value is 1 (ON). |
| Select status text for 0 - OFF | **OFF** | This parameter defines the text in the detail view or in the tile view if the value is 0 (OFF). |

### 8.16. Parameters of the "Dimming Tunable White" function

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Step width | 1...**10**...50% | This parameter defines the percentage by which a step dims. |
| Time between switching and dimming command | 0.1...**0.4**...3.0 s | This parameter defines when switching takes place and when dimming takes place. If the period between pressing and releasing is less than the set value, switching takes place. If it is greater, dimming takes place. |

| | | |
|---|---|---|
| Select status text for 1 - ON | **ON** | This parameter defines the text in the detail view or in the tile view if the value is 1 (ON). |
| Select status text for 0 - OFF | **OFF** | This parameter defines the text in the detail view or in the tile view if the value is 0 (OFF). |
| Minimum colour temperature | 0..**2,000**..10,000 K | This parameter defines the minimum colour temperature that can be set. |
| Maximum colour temperature | 0..**6,500**..10,000 K | This parameter defines the maximum colour temperature that can be set. |

### 8.17. Parameters of the "Blind / shutter step move", "Blind positioning" and "Shutter positioning" functions

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Time between short and long-term command | 0.1...**0.4**...3.0 s | This parameter defines when a short-term and when a long-term command is sent. If the period between pressing and releasing is less than the set value, it is a short-term command. If it is greater, it is a long-term command. |

### 8.18. Parameters of the "Room temperature controller", "Room temperature controller ON/OFF" functions

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Status controller | **Operating mode (20.102 DPT_HVACMode)** | If "Operating mode" is configured, the regular operating mode communication object for receiving the controller status is made available. |
| | Operating status (DPT_HVACStatus) | If "Operating status" is configured, the DPT_HVACStatus that is documented but not adopted into the standard (see KNX documentation) is used as a communication object. |
| Operating mode | **Heating** | Corrected variables of a heating system are used. |
| | Cooling | Corrected variables of a cooling system are used. |
| | Heating and cooling | Mixed mode. A communication object facilitates toggling between the two operating modes. |
| Presence detection with presence button | **active**<br>inactive | Presence detection is carried out through a button on the device or through the presence object (e.g. other touch sensors) as long as the checkmark is activated for this parameter. The presence object is enabled. When the "Presence button" button is pressed, a "1" telegram is transmitted to the KNX via the "Presence object" communication object. |

| | | |
|---|---|---|
| Minimum setpoint | 7.0...**19.0**...40.0 °C | Defines the minimum setpoint temperature. This should be lower than or equal to the lowest other set setpoint value. |
| Maximum setpoint | 7.0...**23.0**...40.0 °C | Defines the maximum setpoint temperature. This should be higher than or equal to the highest other set setpoint value. |
| Step width | 0.1 K<br>0.2 K<br>**0.5 K**<br>1.0 K | This parameter defines the step width in the user interface. |

### 8.19. Parameters of the "Sauna temperature controller", "Sauna temperature controller ON/OFF" functions

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Temperature range | 40-70 °C<br>50-80 °C<br>**60-90 °C**<br>70-100 °C<br>80-110 °C | This parameter defines the temperature range for the sauna. |
| Step width | 1 K<br>**5 K**<br>10 K | This parameter defines the step width in the user interface. |

### 8.20. Parameters of the "Air conditioning" function

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not.<br>Important: Take the structural factors of the AC system into account when using the timers. |
| Minimum setpoint | 7.0...**19.0**...40.0 °C | This parameter defines the minimum setpoint value to be set. |
| Maximum setpoint | 7.0...**23.0**...40.0 °C | This parameter defines the maximum setpoint value to be set. |
| Step width | 0.1 K<br>0.2 K<br>**0.5 K**<br>1.0 K | This parameter defines the step width in the user interface. |

| | | |
|---|---|---|
| Use percentages? | active<br>**inactive** | Communication with actuator via percentage values instead of steps.<br>If this parameter is activated, 1-byte data points of DPT 5.001 are used instead of DPT 5.010. |
| Show automatic operating mode? | **active**<br>inactive | If this parameter is activated, this operating mode is displayed in the user interface. |
| Show heating operating mode? | **active**<br>inactive | If this parameter is activated, this operating mode is displayed in the user interface. |
| Show cooling operating mode? | **active**<br>inactive | If this parameter is activated, this operating mode is displayed in the user interface. |
| Show ventilation operating mode? | **active**<br>inactive | If this parameter is activated, this operating mode is displayed in the user interface. |
| Show dehumidification operating mode? | **active**<br>inactive | If this parameter is activated, this operating mode is displayed in the user interface. |
| Show fan stages | active<br>**inactive** | This parameter allows you to activate configuration of the fan stages. The fan stages are then also displayed in the user interface, and the extension has the option of changing fan stage. |
| Number of fan stages | 1<br>2<br>**3**<br>4<br>5 | This parameter allows you to set the number of fan stages, which permits configuration of additional parameters. |
| Show automatic fan stage? | active<br>**inactive** | This parameter allows you to activate configuration and use of the automatic fan stage. |
| Value for automatic fan stage | **0**...255 | This parameter is used to define value and use in the interface for the automatic fan stage. |
| Show fan stage stop? | active<br>**inactive** | This parameter is used to activate the configuration of value and use in the interface for the fan stage stop. |
| Value for fan stage stop | 0...**6**...255 | This parameter is used to define the value for the fan stage stop. |
| Show maximum fan stage? | active<br>**inactive** | This parameter is used to activate the configuration of value and use in the interface for the maximum fan stage. |
| Value for maximum fan stage | 0...**255** | This parameter is used to define the value for the maximum fan stage. |
| Value for fan stage 1 | 0...**1**...255<br>(0...**25**...100% if "Use percentages?": active) | Configures the value for fan stage 1. (Also in %, depending on setting) |

| | | |
|---|---|---|
| Value for fan stage 2 | 0...**2**...255 (0...**40**...100% if "Use percentages?": active) | Configures the value for fan stage 2. (Also in %, depending on setting) |
| Value for fan stage 3 | 0...**3**...255 (0...**55**...100% if "Use percentages?": active) | Configures the value for fan stage 3. (Also in %, depending on setting) |
| Value for fan stage 4 | 0...**4**...255 (0...**70**...100% if "Use percentages?": active) | Configures the value for fan stage 4. (Also in %, depending on setting) |
| Value for fan stage 5 | 0...**5**...255 (0...**85**...100% if "Use percentages?": active) | Configures the value for fan stage 5. (Also in %, depending on setting) |
| Show horizontal adjustment stop/move? | active **inactive** | This parameter is used to activate the configuration of value and use in the interface for the horizontal adjustment stop and move. |
| Show stages horizontal adjustment? | active **inactive** | This parameter allows you to activate configuration of the horizontal adjustment. The stages of horizontal adjustment are then also displayed in the user interface, and the extension has the option of changing them. |
| Number of horizontal stages | 1...**3**...5 | This parameter allows you to set the number of horizontal stages, which permits configuration of additional parameters. |
| Value for horizontal stop | **0**...255 | This parameter is used to define the value for the horizontal stop. |
| Value for horizontal move | 0...**255** | This parameter is used to define the value for the horizontal move. |
| Value for horizontal stage 1 | 0...**1**...255 (0...**25**...100% if "Use percentages?": active) | Configures the value for horizontal stage 1. (Also in %, depending on setting) |
| Value for horizontal stage 2 | 0...**2**...255 (0...**40**...100% if "Use percentages?": active) | Configures the value for horizontal stage 2. (Also in %, depending on setting) |
| Value for horizontal stage 3 | 0...**3**...255 (0...**55**...100% if "Use percentages?": active) | Configures the value for horizontal stage 3. (Also in %, depending on setting) |

| | | |
|---|---|---|
| Value for horizontal stage 4 | 0...**4**...255<br>(0...**70**...100% if "Use percentages?": active) | Configures the value for horizontal stage 4. (Also in %, depending on setting) |
| Value for horizontal stage 5 | 0...**5**...255<br>(0...**85**...100% if "Use percentages?": active) | Configures the value for horizontal stage 5. (Also in %, depending on setting) |
| Show vertical adjustment stop/move? | active<br>**inactive** | This parameter is used to activate the configuration of value and use in the interface for the vertical adjustment stop and move. |
| Show stages vertical adjustment? | active<br>**inactive** | This parameter allows you to activate configuration of the vertical adjustment. The stages of vertical adjustment are then also displayed in the user interface, and the extension has the option of changing them. |
| Number of vertical stages | 1<br>2<br>**3**<br>4<br>5 | This parameter allows you to set the number of vertical stages, which permits configuration of additional parameters. |
| Value for vertical stop | **0**...255 | This parameter is used to define the value for the vertical stop. |
| Value for vertical move | 0...**255** | This parameter is used to define the value for the vertical move. |
| Value for vertical stage 1 | 0...**1**...255<br>(0...**25**...100% if "Use percentages?": active) | Configures the value for vertical stage 1. (Also in %, depending on setting) |
| Value for vertical stage 2 | 0...**2**...255<br>(0...**40**...100% if "Use percentages?": active) | Configures the value for vertical stage 2. (Also in %, depending on setting) |
| Value for vertical stage 3 | 0...**3**...255<br>(0...**55**...100% if "Use percentages?": active) | Configures the value for vertical stage 3. (Also in %, depending on setting) |
| Value for vertical stage 4 | 0...**4**...255<br>(0...**70**...100% if "Use percentages?": active) | Configures the value for vertical stage 4. (Also in %, depending on setting) |
| Value for vertical stage 5 | 0...**5**...255<br>(0...**85**...100% if "Use percentages?": active) | Configures the value for vertical stage 5. (Also in %, depending on setting) |

| | | |
|---|---|---|
| Display error? | active<br>**inactive** | If this value is active, a 1-bit communication object is provided and display of information that an error exists in the user interface is facilitated. |
| Display error text? | active<br>**inactive** | If this value is active, a 14-byte communication object is provided for text and display of the error text in the user interface is facilitated. |

### 8.21. Parameters when the "Scene extension" function is selected

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Scene number | **1**...64 | According to the KNX standard, objects of data point type 18.001 "Scene Control" can call up or save up to 64 scenes via their number. This is where the scene number to be sent to the KNX is defined. |
| Scene extension with save function | **active**<br>inactive | The function of the scene extension is set here. If the Gira G1 is used as the scene extension, the scenes may be stored in either one or several other KNX devices (e.g. light scene touch sensor).<br>In the case of scene call-up or a save function, the Gira G1 emits a telegram with the relevant scene number via the channel's communication object. |

### 8.22. Parameters of the "Value transmitter" function

| Description | Values | Comment |
|---|---|---|
| Show timer | **active**<br>inactive | This parameter sets whether timers can be configured for this function on the Gira G1 or not. |
| Functionality | **0...100% (1 byte / KNX 5.001)**<br>0...255% (1 byte / KNX 5.004)<br>0...255 (1 byte / KNX 5.010)<br>-128...127 (1 byte / KNX 6.010)<br>Temperature (2 byte / KNX 9.001)<br>0...255 with keypad input (1 byte / KNX 5.010)<br>0...65535 with keypad input (2 byte / KNX 7.001)<br>0...4294967295 with keypad input (4 byte/KNX 12.001)<br>-128...127 with keypad input (1 byte / KNX 6.010)<br>-32768...32767 with keypad input (2 byte / KNX 8.001)<br>-2147483648... 2147483647 with keypad input (4 byte/KNX 13.001)<br>Decimal with keypad input (2 byte / KNX 9.x)<br>Decimal with keypad input (4 byte / KNX 14.x) | This parameter defines the type of the value transmitter. This also means a change in the data point type of the communication objects for this function. |
| Basic value | 0...255 | This parameter defines the basic value. The specified value and the value range are based on the functionality. |
| Minimum value | 0...255 | This parameter defines the minimum value that you can set in the user interface. The specified value and the value range are based on the functionality. |

| | | |
|---|---|---|
| Maximum value | 0...**255** | This parameter defines the maximum value that you can set in the user interface. The specified value and the value range are based on the functionality. |
| Text for unit | **<empty>** | This parameter defines the unit shown in the user interface (e.g. °C for temperature values). |
| Allow value adjustment? | **active**<br>inactive | This parameter defines whether you can change the value via the user interface. |
| Step width | 1<br>2<br>5<br>10<br>20 | This parameter defines the step width in the user interface. |
| Number of decimal places | **0**...2 | This parameter defines the usable decimal places in the user interface. |
| Display status value? | **active**<br>inactive | This parameter defines whether the status is shown in the user interface. |

## 8.23. Parameters of the "Status display" function

| Description | Values | Comment |
|---|---|---|
| Display function | **Status display (1 bit)**<br>Value display 0...100° (1 byte / KNX 5.001)<br>Value display 0...360° (1 byte / KNX 5.003)<br>Value display 0...255% (1 byte / KNX 5.004)<br>Value display 0...255 (1 byte / KNX 5.010)<br>Value display -128...127 (1 byte / KNX 6.010)<br>Value display 0...65535 (2 byte / KNX 7.001)<br>Value display -32768...32767 (2 byte / KNX 8.001)<br>Temperature value display (2 byte / KNX 9.001)<br>Brightness value display (2 byte / KNX 9.004)<br>Value display 0...4294967295 (4 byte / KNX 12.001)<br>Value display -2147483648 ... 2147483647 (4 byte / KNX 13.001)<br>Decimal value display (4 byte / KNX 14.x)<br>Text display (14 byte / KNX 16.000) | This parameter defines the type of the status display. This also means a change in the data point type of the communication objects for this function. |
| Colour of the status displays for I - ON | Red - Green - Blue - **Orange** - Grey (OFF) | This parameter defines the colour of the status LED when the value is I (ON). This parameter is only available for the "Status display (1 bit)". |
| Colour of the status displays for 0 - OFF | Red - Green - Blue - Orange - **Grey (OFF)** | This parameter defines the colour of the status LED when the value is 0 (OFF). This parameter is only available for the "Status display (1 bit)". |

| | | |
|---|---|---|
| Number of decimal places | 0...2 | This parameter defines the usable decimal places in the user interface. |
| Select status text for 1 - ON | **ON** | This parameter defines the text to be shown for 1 (ON) in the user interface. This parameter is only available for the "Status display (1 bit)". |
| Select status text for 0 - OFF | **OFF** | This parameter defines the text to be shown for 0 (OFF) in the user interface. This parameter is only available for the "Status display (1 bit)". |
| Text for unit | **<empty>** | This parameter serves to display the unit in the user interface. This parameter is not available for the "Status display (1 bit)" and for the text display. |
| Additional text | **<empty>** | This text makes it possible to display additional text in the user interface. This parameter is not available for the "Status display (1 bit)". |

### 8.24. Parameters of the "IP camera" function

| Description | Values | Comment |
|---|---|---|
| Video URL | **<empty>** | This parameter defines the URL for an IP camera. It is also possible to specify user details. For example: http://username:password@www.test.de |
| Video codec | **M-JPEG** | The value is fixed and cannot be changed. |

### 8.25. Parameters of the "Open URL" function

| Description | Values | Comment |
|---|---|---|
| URL | **<empty>** | Enter the address of the desired website here.<br>Notes:<br>• Please enter the path, incl. protocol prefix, e.g., "http://..."<br>• Please check the display and functionality of the set up website during start-up because not all websites can be opened via the "Open URL" function. |

## 9. List of icons

| | | | | | |
|---|---|---|---|---|---|
| 1 | Light |  | 23 | Bathroom |  |
| 2 | Sun |  | 24 | Living room |  |
| 3 | Night |  | 25 | Library |  |
| 4 | Favourites |  | 26 | Balcony |  |
| 5 | Door |  | 27 | Bathtub |  |
| 6 | Window |  | 28 | Shower |  |
| 7 | Blinds |  | 29 | Home office |  |
| 8 | Open lock |  | 30 | Bedroom |  |
| 9 | Closed lock |  | 31 | Hotel |  |
| 10 | Open door |  | 32 | Exercise room |  |
| 11 | Heating |  | 33 | Workroom |  |
| 12 | Gas-fired boiler |  | 34 | Garage |  |
| 13 | Gas flame |  | 35 | Loading ramp |  |
| 14 | Temperature |  | 36 | Garden |  |
| 15 | Socket outlet |  | 37 | Flower |  |
| 16 | Dining room |  | 38 | Tool |  |
| 17 | Kitchen |  | 39 | Swimming pool |  |
| 18 | Hallway |  | 40 | Whirlpool |  |
| 19 | Children's room |  | 41 | Sauna |  |
| 20 | Playroom |  | 42 | Staircase |  |
| 21 | Baby change |  | 43 | Poolroom |  |
| 22 | Wine cellar |  | 44 | Laundry |  |

| | | |
|---|---|---|
| 45 | Hot plate |  |
| 46 | Watering |  |
| 47 | Watering can |  |
| 48 | WC Male |  |
| 49 | WC Female |  |
| 50 | WC |  |
| 51 | Heating |  |
| 52 | Cooling |  |
| 53 | Water tap |  |
| 54 | Floor plan |  |
| 55 | Storey |  |
| 56 | Adjoining building |  |
| 57 | Parking deck |  |
| 58 | Car park |  |
| 59 | Cloakroom |  |
| 60 | Conference room |  |
| 61 | Lift |  |
| 62 | Solar collector |  |
| 63 | House |  |
| 64 | Factory |  |
| 65 | Office building |  |
| 66 | Weather station |  |
| 67 | Level-crossing |  |
| 68 | Shopping basket |  |

| | | |
|---|---|---|
| 69 | Alarm |  |
| 70 | Eye |  |
| 71 | Film |  |
| 72 | Music |  |
| 73 | Media |  |
| 74 | Scene |  |
| 75 | Romantic |  |
| 76 | Heart |  |
| 77 | Party |  |
| 78 | Comfort |  |
| 79 | Standby |  |
| 80 | Presentation |  |
| 81 | RGB colour picker |  |
| 82 | Electric iron |  |
| 83 | Forklift |  |
| 84 | Auto |  |
| 85 | Helicopter |  |
| 86 | Camera |  |
| 87 | Emergency exit |  |
| 88 | Escape |  |
| 89 | Holidays |  |
| 90 | Consumption values |  |
| 91 | Diagrams |  |
| 92 | Bell |  |

| | | |
|---|---|---|
| 93 | Clock |  |
| 94 | Timer |  |
| 95 | Calendar |  |
| 96 | Settings |  |
| 97 | Antifreeze |  |
| 98 | Cooling/heating |  |
| 99 | Geothermal energy |  |
| 100 | Tablet |  |
| 101 | TV |  |
| 102 | IT |  |
| 103 | Internet |  |
| 104 | Globe |  |
| 105 | Memory card |  |
| 106 | E-mail |  |
| 107 | User profile |  |
| 108 | Information |  |
| 109 | Save |  |
| 110 | Calculator |  |
| 111 | Dog |  |
| 112 | Cow |  |
| 113 | Warning |  |
| 114 | High rack |  |
| 115 | Message |  |
| 116 | Smoking area |  |
| 117 | Smoking ban |  |
| 118 | Manual function |  |
| 119 | Manual operation |  |
| 120 | Fan |  |
| 121 | Function folder |  |
| 122 | Room functions |  |
| 123 | Watchlist |  |
| 124 | Funnel |  |
| 125 | Magnifying glass |  |
| 126 | Cloudy |  |
| 127 | Rain |  |
| 128 | Eco mode |  |
| 129 | Automation |  |
| 130 | Handset |  |
| 131 | 1-way switch |  |
| 132 | Outdoor area |  |
| 133 | Building part |  |
| 134 | Control cabinet |  |
| 135 | Cellar |  |
| 136 | Ground floor |  |
| 137 | Storey |  |
| 138 | Attic |  |
| 139 | Room |  |
| 140 | Break room |  |

| | | |
|---|---|---|
| 141 | Tearoom |  |
| 142 | Reception |  |
| 143 | Canteen |  |
| 144 | Front door |  |
| 145 | Keywords |  |
| 146 | Terrace |  |
| 147 | Phone |  |
| 148 | Mobile |  |
| 149 | Fax |  |
| 150 | Dot |  |
| 151 | Conservatory |  |
| 152 | Close |  |
| 153 | Reset |  |
| 154 | Plus symbol |  |
| 155 | Link |  |
| 156 | Caps Lock key |  |
| 157 | LED signal light |  |
| 158 | DRA |  |
| 159 | I/O module |  |
| 160 | I/O module input |  |
| 161 | I/O module output |  |
| 162 | Motion detector |  |
| 163 | Motion detector with camera |  |
| 164 | Power supply |  |
| 165 | Remote control |  |
| 166 | Repeater |  |
| 167 | Smoke alarm device |  |
| 168 | Technical detector |  |
| 169 | Door module |  |
| 170 | Glass-breakage sensor |  |
| 171 | Operating unit |  |
| 172 | Alarm control unit |  |
| 173 | Indoor siren |  |
| 174 | Outdoor siren |  |
| 175 | Magnetic contact |  |
| 176 | Hand-held transmitter assault |  |
| 177 | Error |  |
| 178 | Change history |  |
| 179 | Tested, selected |  |
| 180 | Change colour |  |
| 181 | Note |  |
| 182 | Important information |  |
| 183 | Main menu |  |
| 184 | Context menu |  |
| 185 | Change sequence |  |
| 186 | Project scope |  |
| 187 | Rename |  |
| 188 | Delete |  |

| | | |
|---|---|---|
| 189 | Whole page width |  |
| 190 | Navigation arrow |  |
| 191 | Selection / jump to first entry |  |
| 192 | Selection / jump to last entry |  |
| 193 | Selection / forward, play |  |
| 194 | Selection / backward |  |
| 195 | Channel |  |
| 196 | Data point |  |
| 197 | Source |  |
| 198 | Checked, OK |  |
| 199 | Draft |  |
| 200 | Notes |  |
| 201 | Quick |  |
| 202 | Slow |  |
| 203 | Keypad |  |
| 204 | Logic |  |
| 205 | AND |  |
| 206 | Type converter |  |
| 207 | Counter |  |
| 208 | Delay |  |
| 209 | Demultiplexer |  |
| 210 | Block |  |
| 211 | Edge detector |  |
| 212 | Heating/cooling |  |
| 213 | Elapsed operating time meter |  |
| 214 | Hysteresis |  |
| 215 | Multiplexer |  |
| 216 | Inverter |  |
| 217 | Comparator |  |
| 218 | On/off delay |  |
| 219 | OR gate |  |
| 220 | Oscillator |  |
| 221 | PI controller |  |
| 222 | PID controller |  |
| 223 | Random generator |  |
| 224 | Room divider |  |
| 225 | Send-by-change |  |
| 226 | Shading |  |
| 227 | Timer folder |  |
| 228 | Stairway light |  |
| 229 | Value generator |  |
| 230 | XOR |  |
| 231 | Sunrise |  |
| 232 | Press, touch |  |
| 233 | User |  |
| 234 | User group |  |
| 235 | Administrator |  |
| 236 | Installer |  |

| | | |
|---|---|---|
| 237 | Security area 1, main security area |  |
| 238 | Security area 2 |  |
| 239 | Security area 3 |  |
| 240 | Security area 4 |  |
| 241 | Security area 1 multiple |  |
| 242 | Security area 2 multiple |  |
| 243 | Security area 3 multiple |  |
| 244 | Security area 4 multiple |  |
| 245 | I/O module contact open |  |
| 246 | Message / mobile telephone |  |
| 247 | Message / IP, internet |  |
| 248 | Message / telephone |  |
| 249 | Message |  |
| 250 | Message / voice message |  |
| 251 | External active |  |
| 252 | Internal active |  |
| 253 | Internal and external active |  |
| 254 | Alarm |  |
| 255 | Outgoing call |  |
| 256 | External active event |  |
| 257 | Internal active event |  |
| 258 | External active alarm |  |
| 259 | Bell |  |
| 260 | Fire |  |
| 261 | Medical alarm |  |
| 262 | Internal active alarm |  |
| 263 | Alarm forwarding |  |
| 264 | Panic alarm |  |
| 265 | Alerting rule |  |
| 266 | Tamper alarm |  |
| 267 | Supervision alarm |  |
| 268 | Technical alarm |  |
| 269 | Vital monitoring |  |
| 270 | Technical alarm |  |
| 271 | Print |  |
| 272 | Bookmark |  |
| 273 | Page |  |
| 274 | Export document |  |
| 275 | Medal |  |
| 276 | Manual alarm |  |
| 277 | Security guard |  |
| 278 | Device in building |  |
| 279 | Alarm in building |  |
| 280 | Help video |  |
| 281 | Marked corner |  |
| 282 | Alarm system settings |  |

| | | |
|---|---|---|
| 283 | Logic editor |  |
| 284 | Security areas |  |
| 285 | Timers and scenes |  |
| 286 | Visualisation |  |
| 287 | Start-up |  |
| 288 | Help / question |  |
| 289 | Left arrow |  |
| 290 | Right arrow |  |
| 291 | Arrow / redo |  |
| 292 | Arrow / undo |  |
| 293 | Scene set |  |
| 294 | Information, messages |  |
| 295 | Subsections |  |
| 296 | Percent |  |
| 297 | Roof window |  |
| 298 | Server |  |
| 299 | Bluetooth |  |
| 300 | CD |  |
| 301 | Selection / jump to first entry |  |
| 302 | Selection / jump to last entry |  |
| 303 | Input |  |
| 304 | Input jack |  |
| 305 | Lower volume |  |

| | | |
|---|---|---|
| 306 | Network folder |  |
| 307 | MP3 player |  |
| 308 | Radio |  |
| 309 | Speaker |  |
| 310 | User 1 |  |
| 311 | User 2 |  |
| 312 | Action Center |  |
| 313 | Changeover switch |  |
| 314 | NC contact |  |
| 315 | NO contact |  |
| 316 | 12V output |  |
| 317 | 0V output |  |
| 318 | Gira G1 |  |
| 319 | Urgent technical alarm |  |
| 320 | Green tick |  |
| 321 | Query |  |
| 322 | Download |  |