Iddero VERSO 2
VERSO+IP 2
VERSO INDOOR

User manual

Contents

Introduction

1.1 Product overview

Iddero Verso is a family of 4.3" capacitive touch panels designed for visualisation and control of KNX installations. They are available in white or black GLASS finish, and can be installed in portrait or landscape mode.

Function highlights:

- Up to 96 control functions, organized in 12 configurable pages
- User editable favorites page
- Custom background images
- Weekly time schedules: up to 96 channels, 4 programs / channel
- Alarm monitoring: up to 96 alarms, with event log
- Touch gestures: Up to 5 quick actions without leaving power saving mode
- Internal scene controller
- Four independent thermostats
- Four multi-function inputs, individually configurable as binary or temperature probe inputs
- Built-in temperature sensor (not supported in **VERSO** INDOOR)
- Real-time clock (RTC) with backup battery
- Integrated KNX bus coupling unit
- Ultra-low power consumption

The **VERSO+**IP 2 and **VERSO** INDOOR models feature an Ethernet connector, and support remote control from smartphones and tablets through the *Iddero Mobile* app.

The **VERSO** INDOOR model also doubles as an indoor video door phone monitor, and supports IP camera visualisation.

This document uses "Iddero Verso" to collectively refer to all product models, unless stated otherwise.

1.2 Connection

The following figure shows connectors and other elements in the backside of Iddero Verso.

Power supply connector (1)
KNX programming button (2)
KNX programming LED (3)
KNX TP1 bus connector (4)

Multifunction inputs connector

Ethernet connector
USB connector
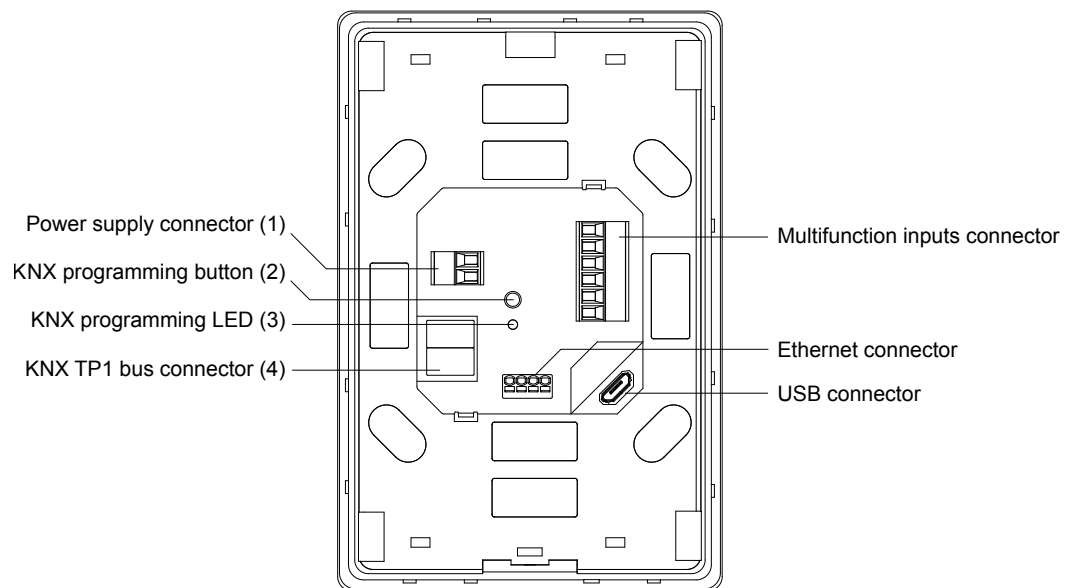
In order to bring up the device, connect the KNX bus to the KNX TP1 connector (4), and a separate power supply (12-30 VDC) to the power supply connector (1). Please refer to the product datasheet for detailed information on the power supply requirements.

> **Note**
>
> Do not connect the KNX bus directly to the power supply connector (1); a separate power supply is required.

1.3 First boot

As soon as the power supply is connected, the device will boot. Within a few seconds, the Iddero logo will appear on the screen. Then, the Synchronisation page will be shown.

Waiting

v1.0

The Synchronisation page will remain onscreen as long as the device has not yet been programmed with application data from ETS. This page is also displayed if the KNX bus is disconnected, or while the device is being reprogrammed.

If the KNX bus is connected and the device has already been programmed, the Synchronisation page will disappear and the Home page will be shown automatically.

1.4 Programming mode

In order to put the device in programming mode, press the KNX programming button (2). While the device is in programming mode, the KNX programming LED (3) will be on.

Functional description

<div style="text-align: right">2</div>

2.1 Visualisation

2.1.1 Overview

The Iddero Verso user interface is organized into pages. This includes both configurable function pages (also called "control pages") and fixed function pages, such as the Menu or settings pages. The following diagram provides a high level view of the navigation structure:
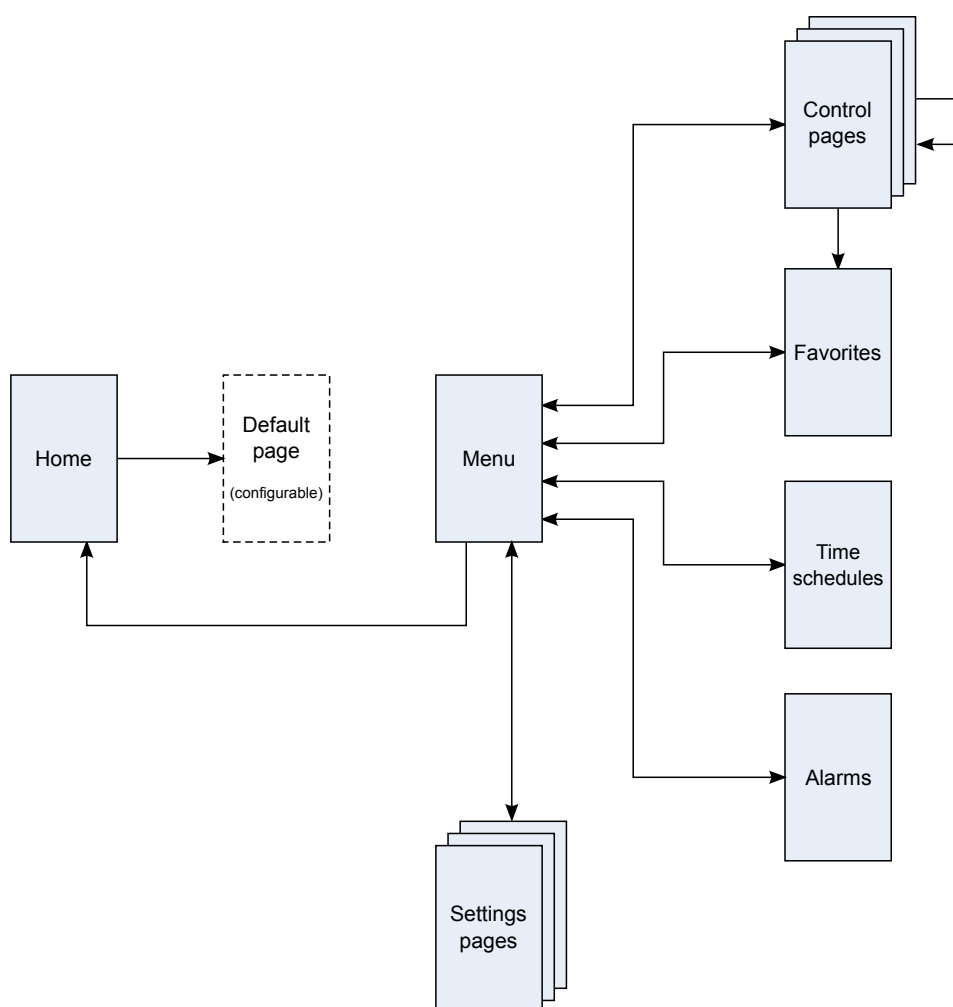
The following screenshots show the main elements of the user interface in both portrait and landscape modes:
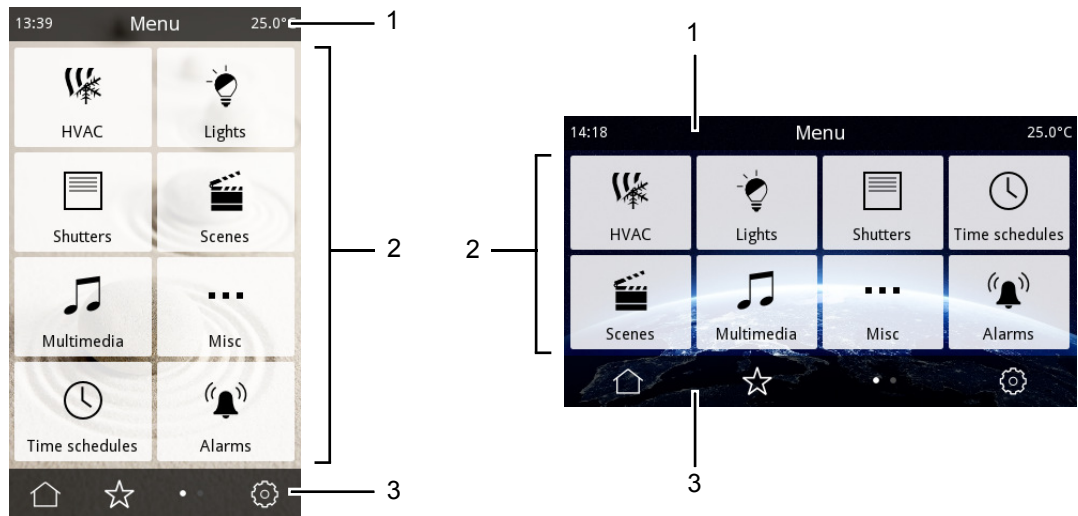
The status bar (1) at the top of the screen shows the name of the current page, and, optionally, the current time and temperature.

The main area of the screen (2) shows the icons, buttons, and user interface elements relative to the current menu or function page.

The navigation bar (3) at the bottom of the screen shows icons to navigate through the user interface. The set of displayed icons will depend on the available navigation options in each page; these are summarized in the following table:

Icon	Function
⌂	Go to the Menu page. If already in the Menu page, exit to Home page.
↩	Return (go back one step)
☆	Go to the Favorites page
⚙	Go to the settings pages
● ○	In the Menu page: switch between available page groups
‹	In control pages: go to previous control page
›	In control pages: go to next control page

2.1.2 Home page
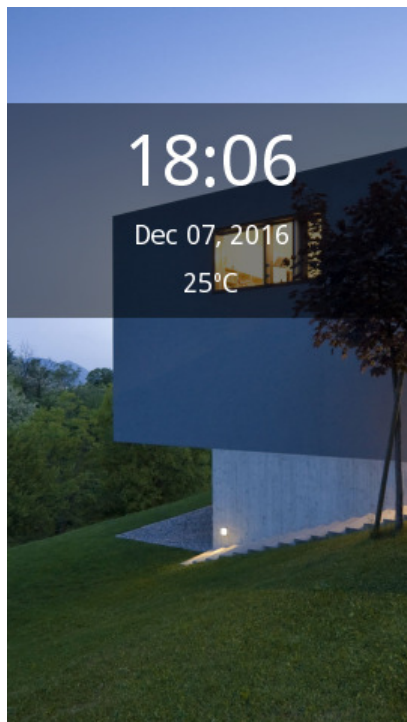
Upon startup, the Home page is shown automatically. The Home page can be configured to show the current date and time, and also the current temperature. Tapping anywhere on the screen will automatically switch to the "default page". The "default page" can be the Menu page, the Favorites page, or any of the configured control pages.

2.1.3 Menu page
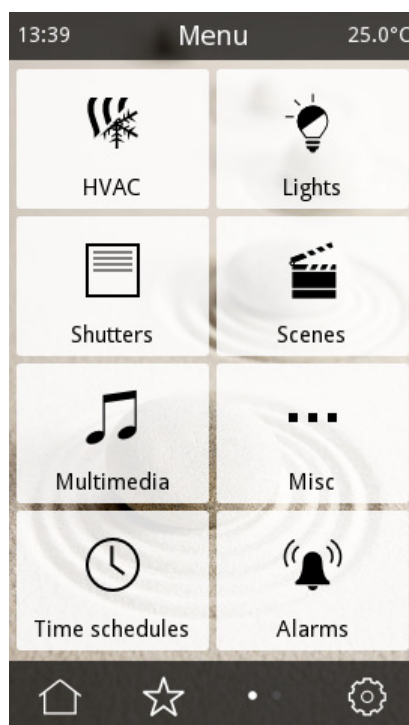
The Menu page provides quick and easy access to all pages in the visualisation, including:

- Any of the configured control pages
- The time schedules pages
- The alarms pages
- The Favorites page
- The settings pages

Control pages are organized in two groups: one for pages 1-6 and another one for pages 7-12. Touch the "dots" icon (• ◦) to switch between page groups.

2.1.4 Control pages
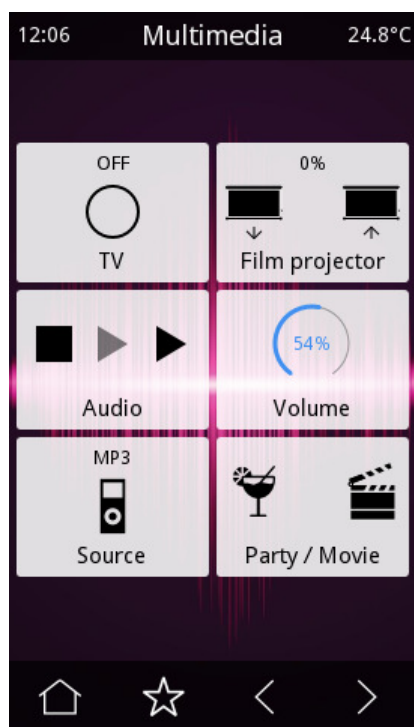
Control pages contain up to 8 configurable widgets, known as "components". Available component types include the following:

- Indicators
- Indicators with alarm function
- Push buttons
- Double push buttons
- Regulation bars
- Rotary controls
- RGB and RGBW controls
- Numeric keypads
- Dimmer controls
- Shutter controls
- Temperature controls
- Door phone page link (**VERSO** INDOOR)
- Cameras page link (**VERSO** INDOOR)

These are described in detail in section 3.4.2 Components on page 56.

For most component types, a timer function can be enabled. See 2.2 Time schedules on page 22.

2.1.5   Favorites page

In addition to the regular control pages, a "favorites" page allows end users to select up to eight commonly used components for quick access. Any of the configured components can be selected; see 2.1.6.4 Edit favorites page on page 17. The Favorites page can be reached from the Menu page or from any of the control pages.

2.1.6   Settings pages

The settings pages allow end users to configure certain device functions, as well as user interface preferences.

2.1.6.1   Main settings menu
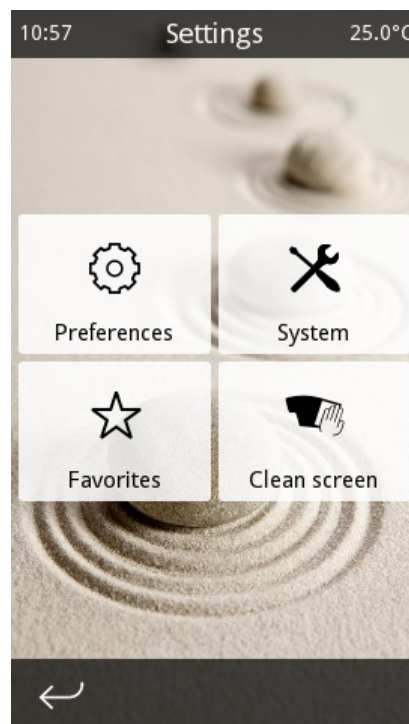
The following buttons are available in this page:

- *Preferences:* Opens the Preferences page. See 2.1.6.2 Preferences page.

- *System:* Opens the System Settings page. See 2.1.6.3 System settings page on page 15.

- *Favorites:* Opens the Edit Favorites page. See 2.1.6.4 Edit favorites page on page 17.

- *Clean screen:* Temporarily disables the touch screen for cleaning. Use a soft cloth to clean the screen. Do not use abrasive cleaners or detergents.

2.1.6.2   Preferences page
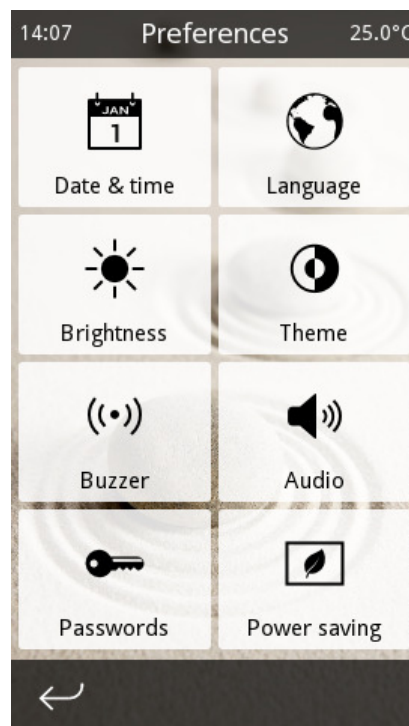
The following buttons are available in this page:

- *Date & time:* Allows setting the system date and time.

- *Language:* Allows selection of the user interface language.

- *Buzzer:* By default, the screen will beep when the user touches a user interface element. The screen also beeps with different tones when a touch gesture is detected, and when an alarm triggers. This menu allows disabling some or all of these sounds.

- *Audio* (**VERSO** INDOOR): Allows adjusting audio levels for the door phone function: volume and mic level during calls, and ring volume.

- *Brightness:* Allows adjusting the brightness of the display.

- *Passwords:* Allows setting the user and/or master passwords. See 2.1.8 Access control on page 20.

- *Power saving:* Allows configuration of the inactivity timeouts used for power saving mode. See 2.1.9 Power saving mode on page 21

- *Theme:* Allows selection between light and dark modes.

2.1.6.3 System settings page
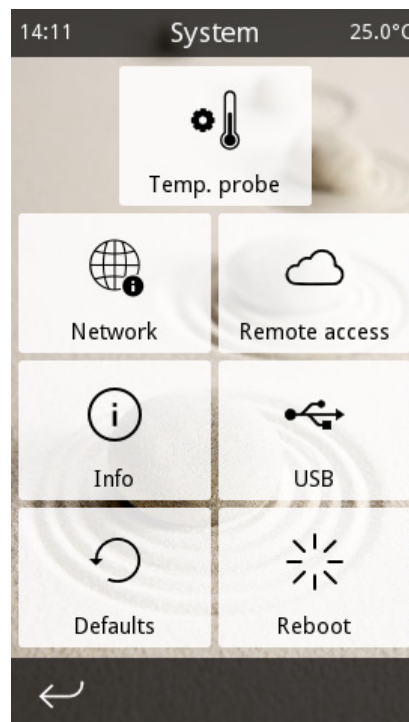
The following buttons are available in this page:

- *Temperature probe:* Allows adjusting the calibration offsets for enabled temperature probes (both the internal temperature sensor and any temperature probe connected to the device's multi-function inputs). The calibration offset is added to the temperature measured by the probe in order to compensate for the effect of local heat sources or air drafts that might affect the measurement.

- *Network* (**VERSO+**IP 2 and **VERSO** INDOOR): Shows the current network settings. Network settings are configured automatically using DHCP for **VERSO+**IP 2, and manually through ETS for **VERSO** INDOOR.

- *Remote access* (**VERSO+**IP 2 and **VERSO** INDOOR): Opens the remote access page, where the touch panel can be linked to the *Iddero Mobile* app for remote control from smartphones and tablets. This page shows the following:

  - The current status of the cloud service
  - The current IP address
  - Buttons to link the device to an idderocloud account, and to unlink the device from the current account (for more information please refer to).

- *Info:* Shows information about the device and the configuration. The following information will be shown:

  - Project name, if parametrized in ETS
  - Firmware and application program version
  - Serial number of the device
  - Current KNX address

- *USB:* Allows advanced customisation options. See .

- *Defaults:* Resets all settings to default values. Any user configuration will be discarded, including preferences, timer schedules, scene values, etc. The device will be left in the same state as if it had been just programmed from ETS. Note: The following settings are not affected by this option:

  - Current date and time
  - Display brightness
  - Audio settings
  - Linked idderocloud account

- *Reboot:* Reboots the device.

2.1.6.4 Edit favorites page
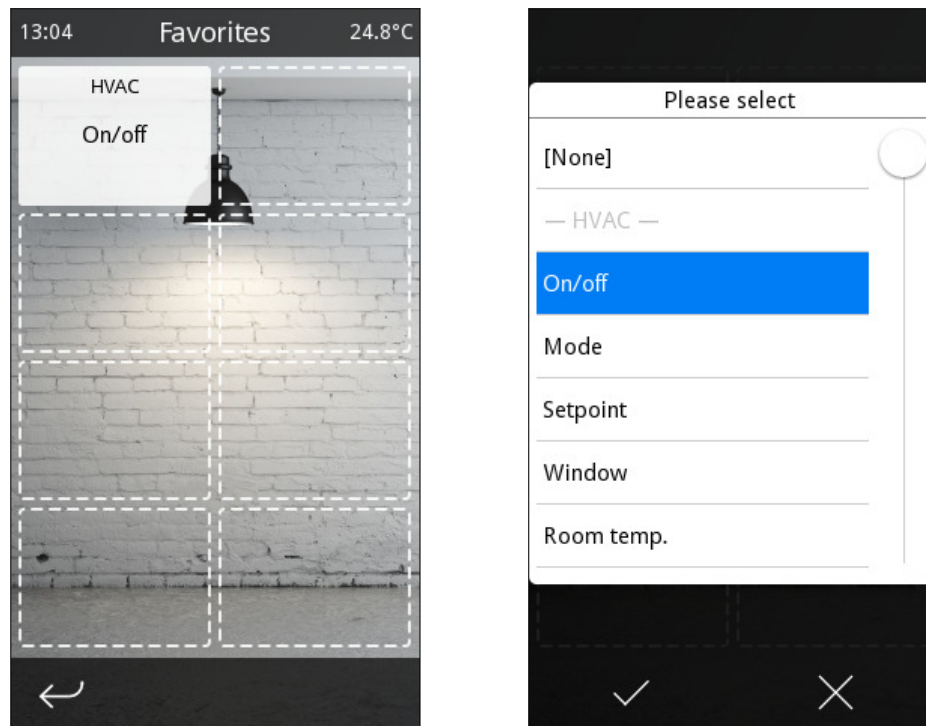
This page allows end users to select up to eight components that will be shown in the Favorites page. A grid with eight positions will be shown. Touching any of the elements in the grid shows a list of all available components. The special list entry "[None]" can also be selected in order to clear the corresponding position in the grid.

2.1.7 Page layout

Many of the pages in the user interface are arranged as a grid that can contain up to 8 elements. The position of each element in the grid is shown in the following figure:
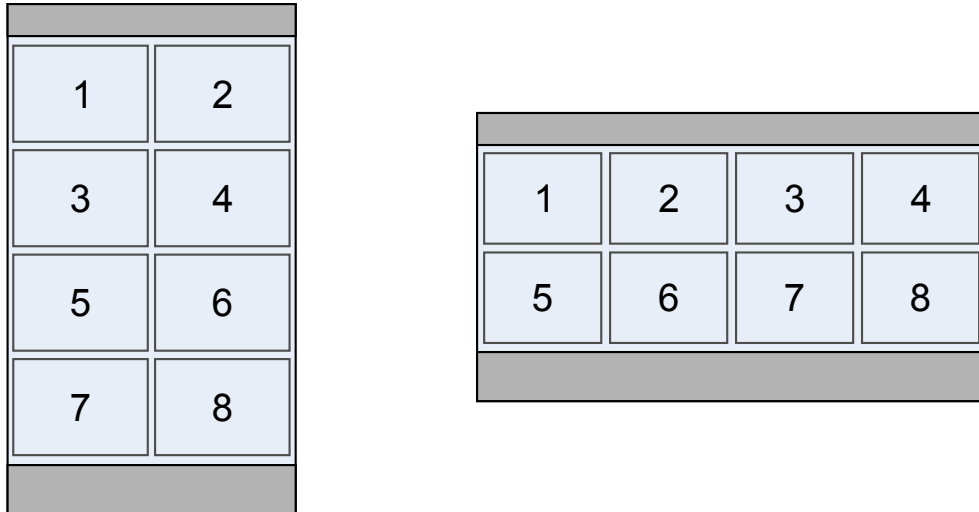
For certain pages, including the Menu page, the configurable control pages, and the Favorites page, several layout options are available.

- *Fixed positions:* Each element is displayed in its natural position in the grid.

- *Flexible – center:* Each row is horizontally centered. Empty rows are skipped; any remaining rows are then vertically centered.

- *Flexible – center + stretch horizontally:* Similar to *flexible – center*, but all rows are horizontally stretched to fill the available screen width.

- *Flexible – center + stretch vertically:* Similar to *flexible – center*, but all rows are vertically stretched to fill the available screen height.

- *Flexible – center + stretch in both directions:* Similar to *flexible – center*, but all rows are horizontally stretched to fill the available screen width, then vertically stretched to fill the available screen height.

The following figure shows the resulting layout for each option in a grid where positions 1, 3, 4, 5, and 6 have been used.
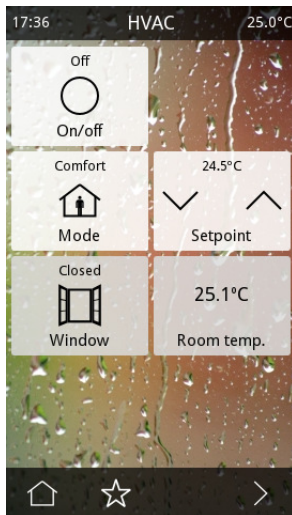
1

2

3
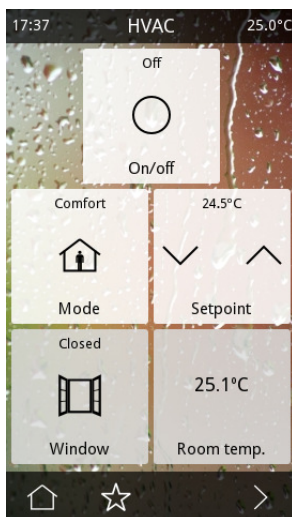
4
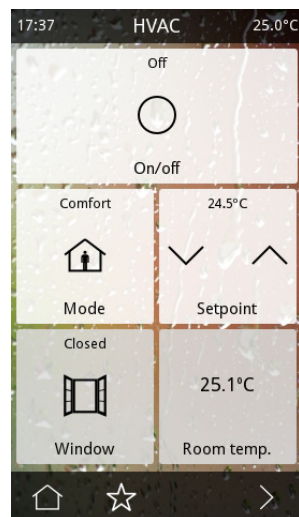
5

1. Fixed positions
2. Flexible – center
3. Flexible – center + stretch horiz.
4. Flexible – center + stretch vert.
5. Flexible – center + stretch in
   both directions

2.1.8 Access control

Most pages in the user interface can be password protected with a 4-digit PIN code in order to prevent unauthorized access.

Two access control schemes can be configured: Either one single password is used for all password-protected pages (one access level), or two different access levels are defined ("user" and "master"). In the latter case, the minimum access level required for each page must be configured. It must be noted that access levels are hierarchical: The "master" password also allows access to pages that require "user" access level. The opposite is not true: The "user" password does not allow access to pages that require "master" access level.

When the user tries to access a protected page, a password dialog will be shown. A valid PIN code must then be entered in order to gain access to the page.
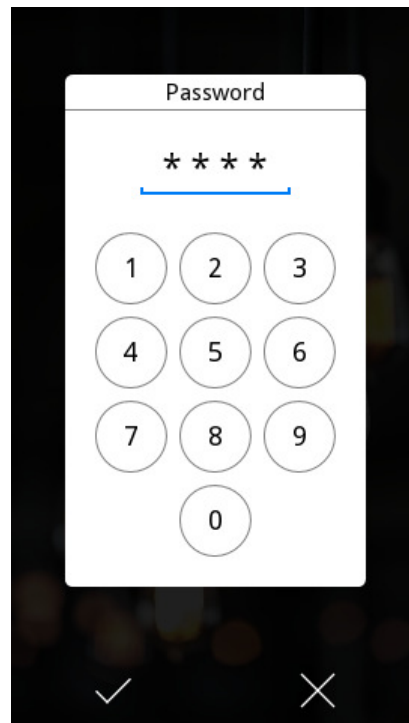
Current permissions remain in effect until the user returns to the Home page, either manually or automatically due to an inactivity timeout (see 2.1.9 Power saving mode on page 21).

2.1.9 Power saving mode

After a certain period of inactivity (no user interaction), the device will automatically return to the Home page. Once on the Home page, and again after a certain period of inactivity, the device enters power saving mode.

During power saving mode, the display is automatically turned off and power consumption is minimized, even though all functions (such as time schedules, alarm monitoring, etc.) remain operational.

Both of these inactivity timeouts can be adjusted by the end user through the "Power saving" option of the Preferences page, or even disabled altogether.

> **Note**
>
> Disabling the power saving timeout is **not recommended** for normal operation, since this may adversely impact the expected display lifetime.

Touching the screen automatically exits power saving mode. It is also possible to configure which page will be shown at this point: Either the Home page or the "default page" defined in the ETS configuration.

2.1.10 Touch gestures

Iddero Verso supports the use of touch gestures for quickly executing up to five preconfigured actions. Touch gestures can be used while the device is in power saving mode, and also in the Home screen. The following gestures are detected:

- Swipe up
- Swipe down
- Swipe left
- Swipe right
- Multitouch gesture (touch the screen with more than one finger)
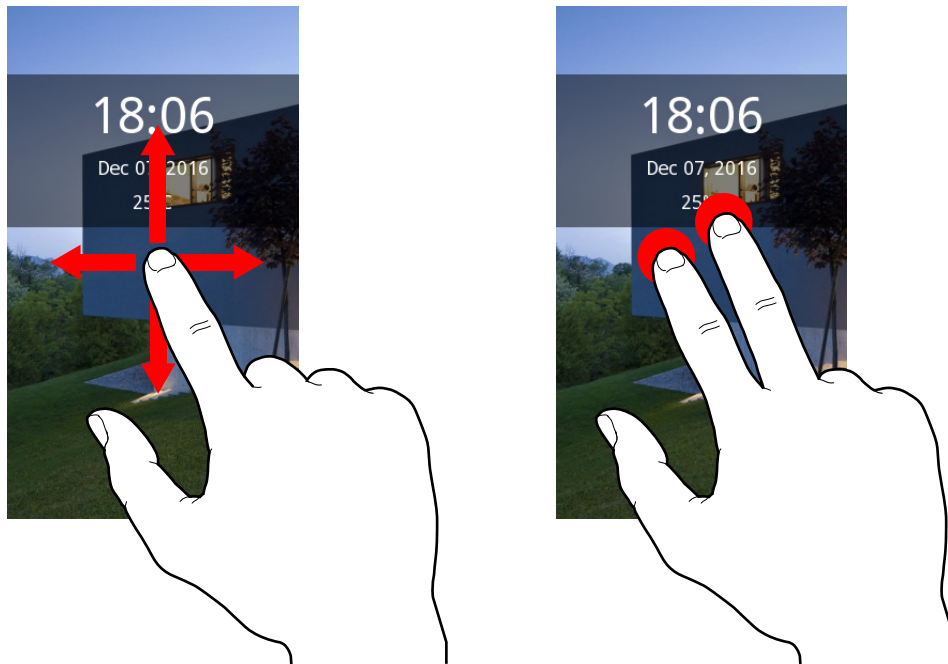
A different action can be configured for each gesture. When the touch gesture is detected, the associated action will be executed and the device will play a short confirmation sound.

Typical uses for touch gestures include switching on the lights on a dark room (the multitouch gesture makes this specially easy, as users just need to place their hand on top of the screen), or activating a "Welcome" or "Good bye" scene when entering or leaving home.

2.2 Time schedules

2.2.1 Overview

For most components in the visualisation it is also possible to enable a "timer function". When the timer function is enabled, the end user can setup actions to be run at scheduled times. For example, a dimmer control can be programmed to switch on a lamp at 75% intensity every day from Monday to Friday at 20:00, and to switch it off at 23:00.

Each component for which the timer function is enabled supports up to four separate time programs, each of which in turn supports separate "start" and "end" actions.

The following parameters can be set by the user for each time program:

- Start and end actions [1]
- Start and end action times
- Week days (SMTWTFS) when the program is valid
- Optionally, a date range (from-to) when the program is valid

The week days parameter can be used to specify which days of the week the program should apply to. The start and end dates make it possible to limit the weekly schedule to a specific period of the year. For example, users could define a program that would only run on weekends during the months of June, July, and August.

2.2.2 Timer groups

Each component enabled as a timer can be optionally added to one of four available "timer groups".

Timer groups can be locked (temporarily disabled) by sending a KNX telegram to a specific communication object. While a timer group is locked, any scheduled actions for components associated to that timer group will not run. This makes it possible to disable a group of timers temporarily, for example during short vacations.

2.2.3 Editing of timer programs

Components for which the timer function has been enabled will show up in the "Time schedules" page, which can be reached from the Menu page.

---

[1] In some cases, actions are fixed and cannot be modified; for example, in a push button component that has been configured to run a scene, the action is fixed (run the associated scene)

Selecting one component from the list shows the available programs:
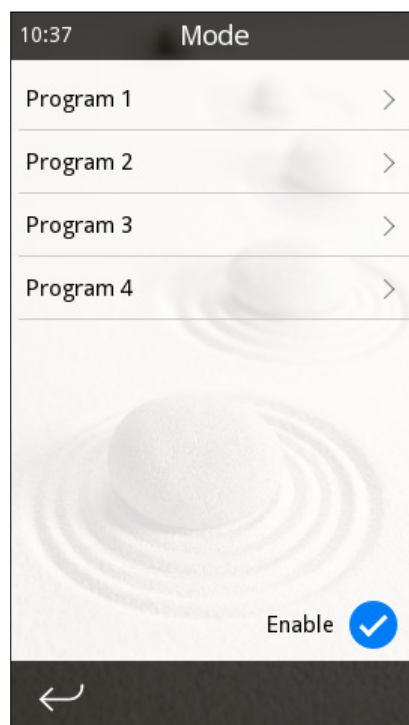
The "Enable" checkbox allows to manually enable or disable any configured actions. Note that this checkbox will be disabled if the component belongs to a timer group that is currently locked.

Selecting a program shows the program edit page, where all the parameters for the timer program can be edited.

When a component has any scheduled actions enabled, a small clock icon ⏱ will be shown both in the component itself and in the corresponding page button in the Menu page.

2.3 Alarms

2.3.1 Overview

Components configured as "Indicator with alarm" combine two functions: On one hand they behave as regular Indicators, and as such they can display status values for an associated communication object. In addition to that, they can also be used to detect alarm or fault conditions.

For this, the component provides two additional communication objects: A "trigger" object, and an "acknowledge" object.

When the "trigger" communication object receives an alarm value from the bus (the "alarm condition"), the alarm is activated. This is signalled both visually, in the user interface, and via the built-in buzzer device. The alarm will remain active until it has been acknowledged by the user and the trigger object has received the "no alarm" value.

Confirmation (acknowledgement) of the alarm can be done either manually, through the user interface, or by writing the acknowledge value to the "acknowledge" communication object.

The following flowchart shows the possible alarm states and the transitions between them:
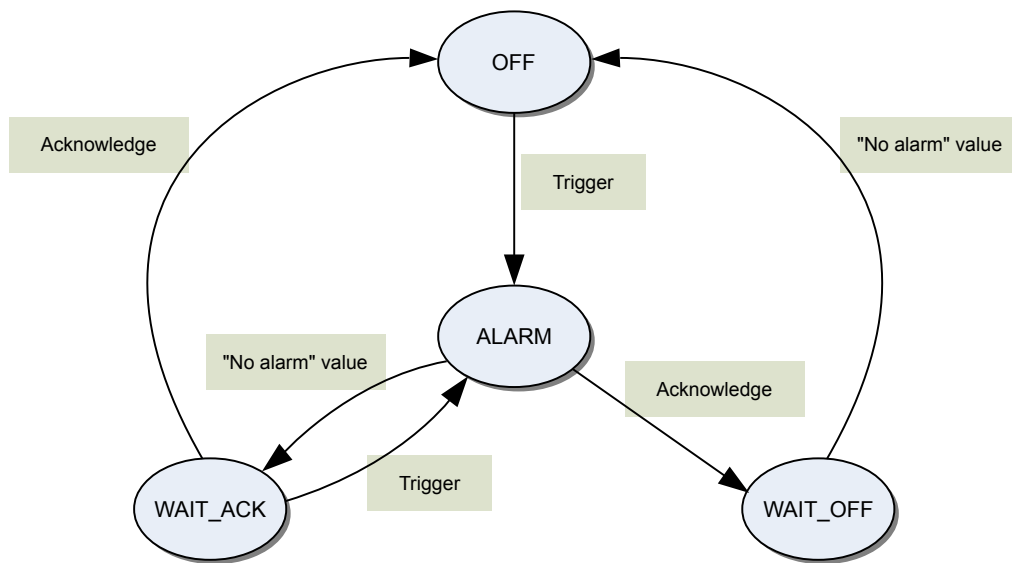
- Initially all alarms are in the OFF state

- When the alarm condition is triggered, the alarm switches to the ALARM state

- If an alarm is in the ALARM state and the alarm condition deactivates (the "no alarm" value is received), it switches to the WAIT_ACK state until it is acknowledged.

- If an alarm is in the ALARM state and the user acknowledges it, it switches to the WAIT_OFF state until the alarm condition deactivates.

- If an alarm is in the WAIT_ACK state and the alarm condition is triggered again, the alarm switches back to the ALARM state.

2.3.2 Alarm visualisation
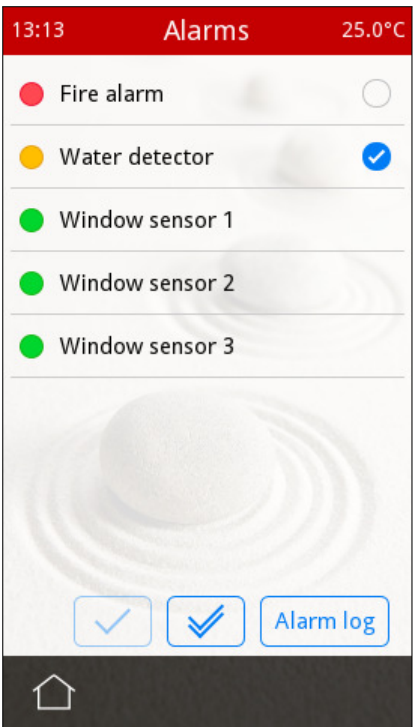
All configured alarms, along with their current state, can be checked at any time from the "Alarms" page, which is reached from the Menu page.

Alarm state is shown as follows:

Symbols	State
🔴 ⚪	ALARM state
🟡 ⚪	WAIT_ACK state (waiting for acknowledgement)
🟡 ✅	WAIT_OFF state (waiting for deactivation of the alarm condition)
🟢	OFF (idle) state

From this page it is also possible to acknowledge individual alarms (by selecting each one and touching the "Acknowledge" (✓) button), or all pending alarms at once with the "Acknowledge all" (✓✓) button.

The alarm status is also shown in the visualisation with a small status led on the component itself. Also, for any page with active alarms, a small status led will be shown in the corresponding page button in the Menu page.

2.3.3 Alarm log
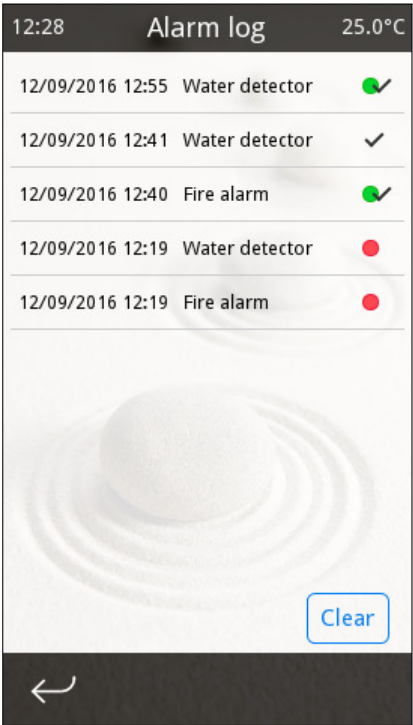
An internal alarm log holds information about the latest 100 alarm events, including the event date and time. The alarm log can be reached from the alarms page ("Alarm log") button. The following events are logged:

Symbol	Meaning
🔴	The alarm is triggered
✓	The alarm is acknowledged, but the alarm condition persists (the "no alarm" value has not been received yet)
🟢✓	The alarm goes idle; it has been acknowledged and the "no alarm" value has been received

The alarm log can be cleared at any time by means of the "Clear" button.

2.4 Thermostats

2.4.1 Overview

Iddero Verso implements four independent thermostats for ambient temperature regulation, and thus can be used to control four independent HVAC zones. Each thermostat can be configured for heating, cooling, or both heating and cooling. The thermostat continuously compares the current room temperature with the current temperature setpoint, and controls operation of the heating and/or cooling actuators using a variety of control methods.

2.4.2 Room temperature

Current room temperature can be obtained from several sources:

- Temperature measured by the internal (built-in) temperature sensor
- Temperature measured by an external temperature probe connected to one of the four available multifunction inputs
- Temperature provided by an external KNX device

For each thermostat, two different temperature sources can be selected and their temperature values averaged in order to calculate the actual room temperature. By combining the measurements from two different sources, it is possible to effectively filter local perturbations, e.g. from heat sources or air streams close to the sensors.

2.4.3 Setpoints

2.4.3.1 Operating Modes

The thermostat has four different operating modes: Comfort, Standby, Economy, and Building Protection.

- **Comfort mode**: The temperature setpoint is set to a value which enables normal use of the room at a pleasant temperature.

- **Standby mode**: The temperature setpoint is temporarily relaxed (i.e. the setpoint will be lowered if in Heating mode, or increased if in Cooling mode), for example during short absences. This helps reducing energy consumption, but at the same time allows for quickly returning to the comfort mode at any time.

- **Economy mode**: If the room will not be used for a longer period of time (for example during the night) the temperature setpoint can be relaxed even further in order to achieve higher energy savings.

- **Building protection mode**: In this mode, the room thermostat is out of service. The room will only be heated (or cooled) if the room temperature has reached values so low (or high) that there is a risk of the installation being damaged from freezing or overheating.

Each of the four operating modes has its own temperature setpoint, as shown in the following diagram:
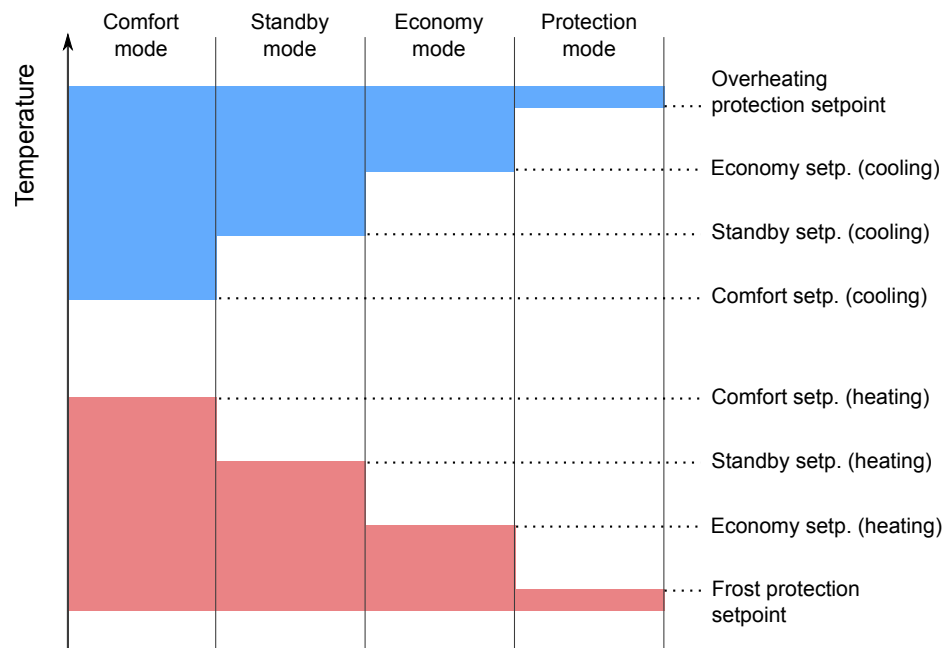
Setpoints can be defined in two ways: Absolute Setpoints and Relative Setpoints. Both methods are described next.

2.4.3.2 Absolute Setpoints

In the Absolute Setpoints method, setpoints for each operating mode are defined as absolute temperature values.

During operation, setpoints can be adjusted at any time by writing to the "Setpoint" communication object. Note that this may trigger an automatic change of operating mode, if the thermostat determines that there is an operating mode that better fits the requested temperature setpoint.

> **Example**
>
> Let's assume the following configuration:
>
> - Thermostat is configured for cooling operation
> - Comfort setpoint is set to 25°C
> - Standby setpoint is set to 27°C
>
> If the thermostat is initially in Comfort mode, and a value of 28°C is written to the "Setpoint" communication object, it will automatically switch to Standby mode.

A configuration parameter controls if and when temperature setpoint changes are stored permanently:

- **Never**: Temperature setpoint changes are never stored permanently.

- **Manually**: A "Save setpoint" communication object is enabled. By writing to this object, the current setpoint value will be stored as the default setpoint for the currently active operating mode, overwriting the initial value that was set in ETS.

- **Upon mode changes**: The current setpoint value will be automatically stored as the default setpoint for the currently active operating mode whenever there is an explicit change of operating mode, or when the thermostat switches from heating to cooling or vice-versa[2].

Setpoints can be restored to their original values (as defined via ETS parameters) at any time via the "Reset setpoints" communication object.

The above discussion applies to the Comfort, Standby, and Economy operating modes. It is not possible to manually adjust the setpoint for Building protection mode.

2.4.3.3 Relative Setpoints

In the "Relative Setpoints" method, the effective setpoints for the Comfort, Standby, and Economy operating modes are defined relative to a common base value (the **base setpoint**), according to the following equation:

Effective setpoint = Base setpoint + Mode offset + User offset

---

[2] Additional considerations apply if the thermostat is configured for automatic switching between heating and cooling. See Additional considerations if Absolute Setpoints are used on page 35.

The base setpoint can be modified at any time through the "Base setpoint" communication object. The **mode offset** for each mode is configured via ETS parameters, and cannot be changed at runtime. The **user offset** allows additional adjustment of the current setpoint within a configurable range.

The user offset can be modified through two communication objects: Either directly writing an offset value to the "Setpoint offset" object, or by writing a "1" or "0" value to the "Setpoint step" object. The latter will result in progressive adjustments of +/- 0.5ºC in the current user offset.

A configuration parameter, "Reset user offset on operating mode changes" determines whether the current user offset should be reset to zero or carried over when the operating mode changes.

Finally, an additional "Reset offset" communication object allows to reset the user offset to zero at any time.

The above discussion applies to the Comfort, Standby, and Economy operating modes. The setpoint for the Building protection mode is always defined as an absolute setpoint, and cannot be changed at runtime.

2.4.3.4 Switching of operating modes

It is possible to toggle between operating modes at any time by writing to the associated communication objects. This can be done via a 1-byte object or via four 1-bit objects, one per mode. In turn, the 1-bit objects can be configured to work in "switch mode" or in "trigger mode":

- **Trigger mode**: Writing a value "1" to one of the 1-bit objects will trigger the activation of the corresponding operating mode.

- **Switch mode**: Writing a value of "1" or "0" to any of the 1-bit of the objects will update the operating mode. The new operating mode will depend on the current value of all four 1-bit switch objects. The possible combinations are shown in the following table ("-" means that the object value is ignored).

Protection object	Comfort object	Economy object	Selected mode
1	-	-	Protection
0	1	-	Comfort
0	0	1	Economy
0	0	0	Standby

> **Note**
>
> Writing to the 1-byte mode object will always set the operating mode regardless of the value of the 1-bit objects, and vice-versa.

When a new operating mode is selected, the current temperature setpoint will be set to that of the new operating mode.

2.4.3.5 Forced protection mode

Under certain circumstances, it may be desirable to override the currently selected operating mode, and force the thermostat into Protection mode. This is called "Forced protection" mode.

This could be the case, for example, when windows are opened in an hotel room. For this purpose, an additional "Window status" communication object can be enabled, which could be connected to e.g. a window sensor. When the activation value ("1") is written to this object, the thermostat automatically switches into Forced protection mode, and will remain in this mode as long as the communication object does not receive the deactivation value ("0").

If the thermostat receives a request to change operating mode while it is in Forced protection mode, the request will be processed, but the change will not be effective until Forced protection mode is deactivated.

2.4.4 Heating and Cooling

Each thermostat can be configured to operate in heating mode only, in cooling mode only, or in combined heating and cooling mode.

If combined heating and cooling operation is selected, switching between heating and cooling can be done either manually (by means of an associated communication object) or automatically.

2.4.4.1 Automatic switching

If automatic switching of heating/cooling is selected, the thermostat will automatically determine the heating/cooling mode depending on the current room temperature and the configured temperature setpoints.

- If the thermostat is in **cooling mode**, it will automatically switch to Heating if the room temperature falls below the corresponding setpoint[3] for Heating mode.

- If the thermostat is in **heating mode**, it will automatically switch to Cooling if the room temperature rises above the corresponding setpoint for Cooling mode.

> **Example**
>
> If the thermostat is in heating / standby mode, it will automatically switch to cooling when the room temperature rises above the setpoint configured for cooling / standby mode.

Changeover protection band

In order to avoid situations where the thermostat is continuously switching between heating and cooling, an additional parameter is introduced that allows defining a "Changeover protection band". The thermostat will not switch between heating and cooling until the difference between the room temperature and the current setpoint is *greater* than the protection band.

> **Example**
>
> Let's assume the following configuration:
>
> - Thermostat is currently in Cooling mode
> - Current operating mode is Comfort
> - Comfort setpoint for cooling is set to 25ºC
> - Comfort setpoint for heating is set to 24ºC
> - Changeover protection band is set to 2ºC
> - Initial room temperature is 26ºC
>
> Normally, the thermostat would switch to heating mode as soon as the room temperature falls below 24ºC. However, because the changeover protection band parameter is set to 2ºC, the switch to heating mode will only happen when the room temperature falls below 23ºC.

---

[3] The setpoint that is considered is the one corresponding to the same operating mode that is currently active (comfort, standby, economy, protection) but for Heating mode.

Additional considerations if Absolute Setpoints are used

In order to avoid malfunction of the thermostat, if the Absolute Setpoints method is used with automatic switching between heating and cooling, an additional limitation is introduced regarding permanent storage of setpoint changes. Namely, the device will always ensure that the comfort setpoint for cooling mode is never lower than the comfort setpoint for heating mode.

2.4.5 Control algorithms

The main function of the Iddero Verso thermostats is to compare the current room temperature with the desired temperature (setpoint) and send appropriate commands to the KNX actuators (relay outputs, valve motors, etc.) that control the actual HVAC equipment.

To this end, Iddero Verso thermostats support different control algorithms:

- Two-point control with hysteresis
- P-I control with continuous (analogue) output
- P-I control with PWM (switched) output

These algorithms are described next.

> **Note**
>
> If a thermostat is configured for both heating and cooling, a different control algorithm can be configured for each function. Thus, for example, a thermostat can be setup to use two-point control with hysteresis for cooling, and P-I control with PWM output for heating.

2.4.5.1 Two-point Control with Hysteresis

Two-point control with hysteresis is a simple and well-known control algorithm that is sometimes also referred to as "On/off control". In this algorithm, the thermostat turns on the heating when the current room temperature is below the desired setpoint, and turns it off when the room temperature rises above the desired setpoint. The opposite holds true for cooling.

In order to prevent the output from quickly switching on and off when the room temperature is close to the setpoint value, different thresholds are considered for switching on and off. This behaviour is controlled with the *Hysteresis* configuration parameters (upper and lower hysteresis).

In Heating mode:

- The output is switched on when the room temperature is less than or equal to *Setpoint - lower hysteresis*
- The output is switched off when the room temperature is greater than or equal to *Setpoint + upper hysteresis*
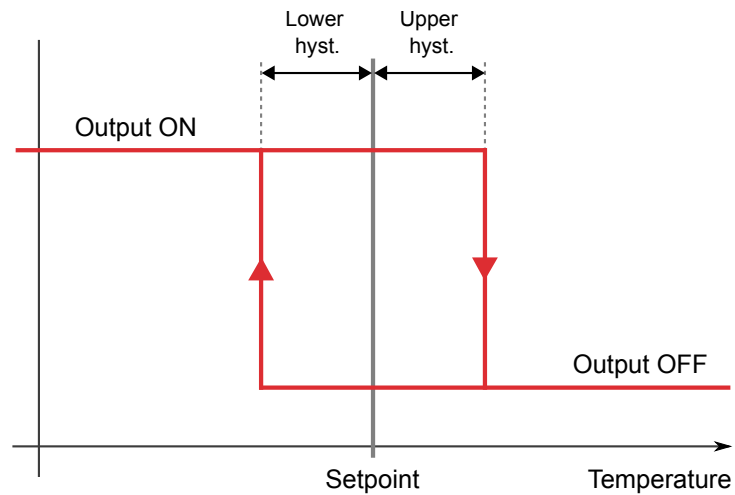
In Cooling mode:

- The output is switched on when the room temperature is greater than or equal to *Setpoint + upper hysteresis*
- The output is switched off when the room temperature is less than or equal to *Setpoint - lower hysteresis*
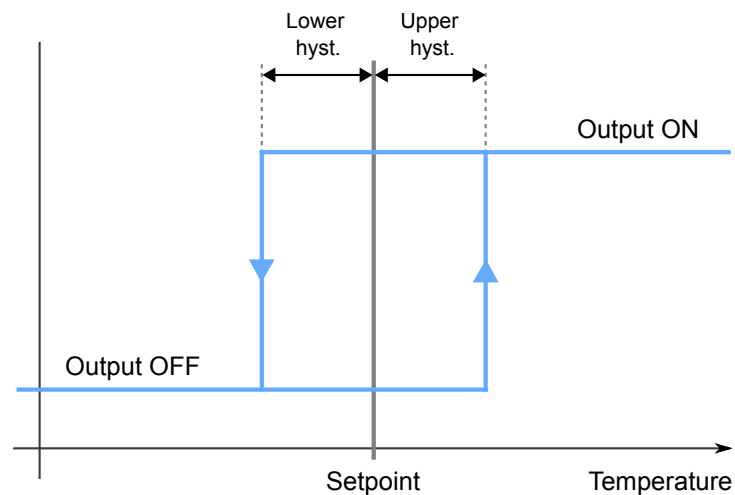
Two-point control with hysteresis is easy to setup and can be used with regular switching actuators. However, it is also sub-optimal in terms of power consumption and comfort, due to the typical fluctuations of the room temperature around the setpoint which arise as a result of the simple control algorithm.
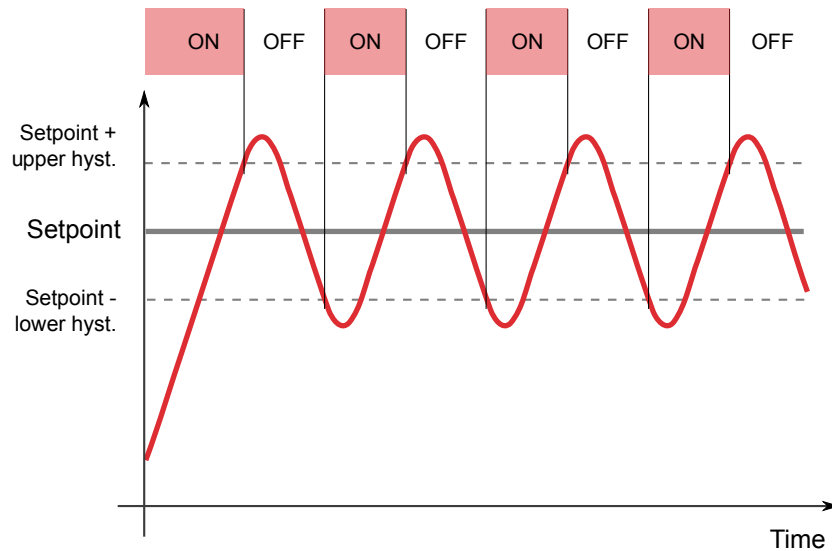
2.4.5.2 P-I Control with Continuous Output

Proportional-Integral (P-I) Control is a more complex control algorithm where the thermostat constantly measures the difference between the current room temperature and the desired setpoint (the *error term*), and computes an appropriate *correction value* based on the current difference, and also on the previous history of the system. This correction value (e.g. 0-100%) is periodically sent to a proportional HVAC actuator device.

When configuring a thermostat to use P-I Control, the following parameters must be considered:

- **Cycle time:** The control algorithm periodically compares the current room temperature with the desired setpoint, and computes the correction values to be applied. The cycle time determines how often this is done. Longer cycle times (typically 15-20 minutes) should be chosen for HVAC systems with high thermal inertia, such as radiant floor heating, and shorter cycle times (typically 10-15 minutes) for HVAC systems with low thermal inertia, such as air convection heating and cooling.

- **Proportional band:** The proportional band (measured in degrees) determines the magnitude of the output value relative to the error term. For example, a proportional band of 5°C means that the output will be set to 100% when the error (the difference between the

current temperature and the desired setpoint) reaches 5°C. If the difference is only 1°C (20% of the proportional band), the output will be set to 20%, and so on.

- **Integral time:** The P-I control algorithm not only considers the current value of the error term at any given time, but also the "history" of the system; that is, the past values of the error term. This makes it possible to react faster to changes in ambient conditions that impact the room temperature. Short integral times allow for fast reactions to such changes, but a too short integral time introduces a risk of permanent oscillations around the setpoint. Long integral times eliminate the risk of oscillations, but the reaction to external changes will be slower.

> **Note**
>
> Every time the desired temperature setpoint changes, the current cycle is interrupted, the new values are processed, and a new cycle is started. This allows the system to reach its steady state more quickly.

If the P-I control parameters are properly adjusted, the control algorithm may be able to efficiently control all common types of heating and cooling systems, thus making the room temperature control work as fast as possible and without deviation. However, tuning of the P-I control parameters is not straightforward, and even small changes to the parameters values can result in noticeable changes to the control performance: An incorrectly configured system may be too slow, needing a long time to reach the desired temperature setpoint, or it may fluctuate above or below the setpoint.

For this reason, a set of *predefined parameters* that correspond to the most common types of heating and cooling systems are available and can be easily selected, as shown in the following table. For expert users, the option of entering custom parameter values is also provided.

Application (heating)	Proportional band (K)	Integral time (min)	Recommended cycle time (min)
Warm water heating	5	150	15
Floor heating	5	240	15-20
Electric heating	4	100	10-15
Convection heating	4	90	10-15

Application (cooling)	Proportional band (K)	Integral time (min)	Recommended cycle time (min)
Cooling ceiling	4	240	15-20
Convection cooling	4	90	10-15

2.4.5.3 P-I Control with PWM Output

In the P-I Control algorithm described in the previous section (P-I Control with Continuous Output) the control variable that is periodically sent to the HVAC actuator is an analogue variable (0-100%). Thus, a proportional HVAC actuator is required.

In some cases, however, it is desirable to benefit from the advantages of P-I control, while still being able to use standard (lower cost) switching actuators, such as relays and actuators for zone valves. For this, the P-I Control algorithm with PWM Output can be used.

This is exactly the same as P-I Control with continuous output, except that proportional output values are implemented by switching a binary output on and off for certain time periods. The control algorithm operates periodically over a cycle (the cycle time); in each period, the output is switched on for a time that is proportional to the value of the control variable, and then switched off for the remaining of the period. For example, if the cycle time is 15 minutes, and the control variable should be set to 20%, the controller will keep the output on for 3 minutes (20% of the cycle time), and off for 12 minutes (80% of the cycle time).

By varying the ratio between the on and off times, also known as the *duty cycle*, the average time of activation of the output, and in turn the average supply of heating or cooling power, can be effectively controlled.
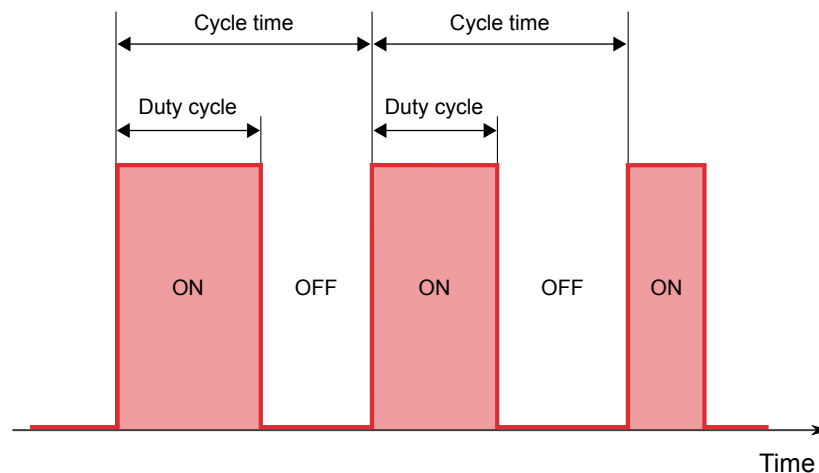
> **Note**
>
> In order to avoid damage to the switching actuator, if the computed on or off time is shorter than 5 seconds, a fixed time of 5 seconds is used instead.

2.4.6 Additional heating and cooling

Some HVAC systems, such as radiant floor heating systems, exhibit a very high thermal inertia, which translates to slow response times at start-up, or when there are changes in the working conditions (ambient temperature or desired setpoint).

In order to improve response times for such systems, auxiliary heating or cooling systems with lower thermal inertia are often installed. These auxiliary systems, often called "second stage" systems, are temporarily switched on when there is a large difference between the room temperature and the setpoint, thus contributing to a faster heating or cooling of the room, and are switched off when this difference becomes smaller and can be handled by the main system.

The Iddero Verso thermostats can be configured to control these auxiliary heating or cooling systems, if they are installed. For this, it is necessary to define a temperature range where the auxiliary system will kick in. This is configured through the "Additional heating (or cooling) band" parameter:

In Heating mode:

- Additional heating is switched on when the room temperature is less than or equal to
  Setpoint - additional heating band
- Additional heating is switched off when the room temperature is greater than or equal to
  Setpoint - additional heating band + 0.5 ºC

In Cooling mode:

- Additional cooling is switched on when the room temperature is greater than or equal to
  Setpoint + additional cooling band
- Additional cooling is switched off when the room temperature is less than or equal to
  Setpoint + additional cooling band - 0.5 ºC

2.5 Multifunction inputs

Iddero Verso features four multifunction inputs, each of which can be individually configured in one of the following ways:

- Binary - Push button
- Binary - Switch/sensor
- Temperature probe

For a detailed description of the available configuration options for each type of input, see 3.6

2.5.1 Binary inputs

2.5.1.1 Push button inputs

Push button inputs are typically connected to external push buttons, and can be configured to trigger different actions upon detection of "short" and "long" presses. For example, a push button input can be connected to a physical push button to control a light dimmer. Short presses can be configured to toggle the light on and off, while a long press can be configured to start a dimming operation.

For each input configured as a push button input, a configuration parameter allows to customize the minimum press time to differentiate between short and long press operations.

2.5.1.2 Switch/sensor inputs

Switch/sensor inputs are typically connected to external switches or sensors, and can be configured to trigger different actions when the input contact closes and when the contact opens.

A typical use case for switch/sensor inputs is to connect external alarm sensors; for example, a fire detector, or a magnetic door/window sensor. Different actions can be triggered when the alarm condition is detected, and also when the alarm condition deactivates.

2.5.1.3 Input locking

For each binary input, an optional "Input lock" communication object is available. This object allows to temporarily disable operation of the associated input. While an input is locked, short and long presses (for push button inputs) or state transitions (for switch/sensor inputs) are ignored, and the corresponding actions are not triggered.

2.5.2 Temperature probe inputs

Temperature probe inputs can be used to measure temperature from connected temperature probes.

A configurable offset can be added to the measured value in order to compensate for the effect of local heat sources or air drafts that might be close to the probe location.

Temperature values are made available through an associated communication object. Values can be sent to the bus periodically, and/or when a certain change between the current value and the last value sent to the bus is detected. If the temperature cannot be read (for example, if the probe is broken, or if it is physically disconnected from the input) an error condition is signalled through a separate error object.

An **alarm function** can be enabled for each temperature probe input. The alarm function can be configured to detect high temperatures, low temperatures, or both high and low temperatures. When the temperature value exceeds the configured limits, an alarm condition is signalled through a specific "alarm" communication object.

2.6 Scene controller

2.6.1 Internal vs. External scenes

Many of the functions in Iddero Verso (such as touch gestures, or components used in the visualisation) can be configured to trigger so called "external" scenes. External scenes are managed by external KNX devices, which are in charge of storing the scene values, and recalling them when the scene is triggered. In this case, Iddero Verso simply sends a KNX telegram indicating the scene to be recalled or stored.

In addition to this, Iddero Verso features an internal scene controller. "Internal" scenes are managed internally, and can be recalled or stored in response to internal actions, and also if requested by external KNX devices.

2.6.2 Scene actuators

Up to 8 scene actuators can be enabled. Each actuator is associated with a communication object, and can be configured to respond to up to 8 different scenes (which need not be the same for all actuators).

Scenes can be recalled and stored in a variety of ways:

- When the user activates a scene push button or double push button component in the visualisation
- Through the "Run/save internal scene" communication object
- As a response to certain actions, such as touch gestures

2.6.3 Recalling scenes

When the internal scene controller receives a request to recall a given scene, all scene actuators are checked. For each scene actuator that has been configured to respond to the corresponding scene number, the value associated with that scene number will be sent to the bus.

2.6.4 Storing scenes

When the internal scene controller receives a request to store a given scene, all scene actuators are checked. For each scene actuator that has been configured to respond to the corresponding scene number, the current value of the associated communication object will be stored as the new value for that scene number (as long as the parameter "Allow saving" is enabled for that scene).

The following must also be considered:

1. If the associated communication object has not been updated since startup (either by the device itself or by reception of a value from the bus), then the actuator will be skipped, and the value associated with the corresponding scene number will remain unchanged.

2. If the parameter *Send read requests at startup* is enabled, the device will send read requests for all scene actuators' communication objects at startup. Values received as a response to these read requests will be used as default values to store when a "store request" is processed, if no other value has been written to the corresponding communication object by that time.

2.7 Door phone function

Iddero **VERSO** INDOOR can be used as a modern indoor monitor capable of receiving calls from compatible SIP-based door entry panels. Up to 8 door stations (outdoor units) can be configured; for each one it is possible to control one main door and one secondary door.

2.7.1 Incoming calls

When an incoming call is received (that is, when a guest presses the call button in the door phone station) the touch panel will play a ring tone through the integrated loudspeakers, and the status bar at the top of the screen will turn green.

If the touch panel was idle (in power saving mode, or in the Home page), it will automatically turn on and display the Door phone page.

If the touch panel is in use when an incoming call is received (i.e. it is turned on and in any page other than the Home page), the Door phone page is not automatically displayed. In this case, the user must navigate to the Door phone page manually, as described in the following section.

2.7.2 Manual navigation to the Door phone page

In order to navigate to the Door phone page, at least one component of type "Door phone page link" must be configured in the visualisation (see 3.4.2 Components on page 56).

The user can navigate to the Door phone page at any time, even when there is no incoming call. If there is no incoming call, depending on the actual configuration it may be possible to watch the video stream from the door station and open the doors. However it is not be possible to establish audio communication.

2.7.3 The Door phone page
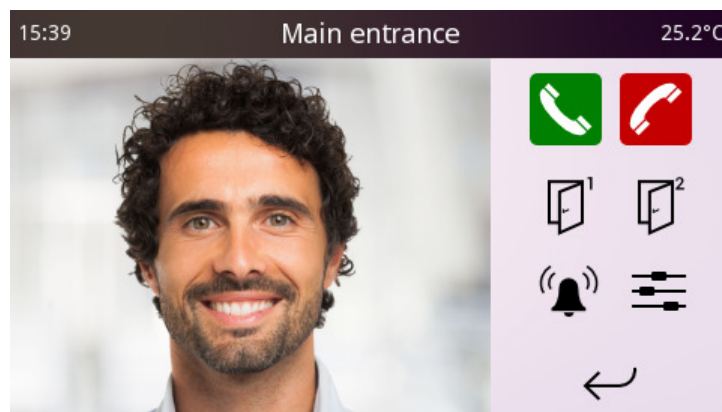
When entering the Door phone page, the video from the currently selected door station will be displayed. If multiple door stations are enabled, you can switch between door stations by touching the image.

The following table describes the function of each button in the user interface:

Icon	Function
📞	Accept an incoming call and establish audio communication
📞	Reject an incoming call / terminate call in progress
🚪¹	Send the door open signal for the main door (the icon is configurable)
🚪²	Send the door open signal for the secondary door (the icon is configurable)
🔔	Open the ring volume settings dialog
☰	Open the in-call audio settings (volume and mic level during calls)

If multiple door stations are enabled, please keep in mind the following:

- The pick up / hang up buttons always relate to the call currently in progress, regardless of which door station is currently displayed.

- The door opening buttons always relate to the door station currently displayed. They may or may not be shown depending on configuration.

- Audio settings are global and affect communications with all door stations.

2.8 Cameras

Iddero **VERSO** INDOOR supports visualisation of IP cameras.

In order to enable the Cameras page, at least one component of type "Cameras page link" must be configured in the visualisation (see 3.4.2 Components on page 56).

The Cameras page shows a list of all cameras that have been configured in the project. Select a camera from the list to view the live video captured by the camera.

If the touch panel is configured in portrait mode, the video from the camera will be displayed in the same page. If the touch panel is configured in landscape mode, the video from the camera is displayed in full screen mode. Touch anywhere on the screen to go back to the cameras list.

Configuration 3

3.1 General considerations

This chapter describes the configuration parameters available in the ETS product database for **VERSO** 2, **VERSO+**IP 2 and **VERSO** INDOOR.

Configuration parameters are organized in groups, or *configuration sections*; this chapter follows the same structure. For each section, the corresponding configuration parameters are shown in table format. For each parameter, available options are listed. The default value is indicated in **bold**.

Some parameters allow entering text values. These are labeled as [Text field] in parameter tables. Regular text fields have a maximum length of 20 bytes. Text fields used for units in numerical values have a maximum length of 6 bytes. In some cases, the maximum length might be different.

> **Note**
>
> All texts are stored using Unicode UTF-8 encoding. Encoding of ASCII characters requires 1 byte per character; however, encoding of characters outside the ASCII range requires more than 1 byte per character. Thus the maximum number of characters allowed will depend on the actual characters used in the text. It is recommended to check that the full texts are actually visible on the screen after programming.

Some parameters allow selecting an icon. These are labeled as [Icon selection]. For a complete list of available icons, please refer to the Iddero Verso icon list, available for download from www.iddero.com.

In most cases, when an icon is used, the icon colour can also be configured. These parameters are labeled as [Icon colour] in parameter tables.

Some parameters are related to the touch panel's built-in temperature sensor. This sensor is not supported initially in **VERSO** INDOOR; thus, the corresponding parameters or configuration options are not available in the application program.

3.2 Main

3.2.1 General

Parameter name	Values
Project name	[Text field]
Display orientation	**Portrait** / Landscape
Enable date/time objects	[Checkbox]
Date/time sending period *(If date/time objects are enabled)*	**0** ... 255 (x 1 min, 0 = Disabled)
Send read requests at startup	**No** / Yes
Startup requests initial delay *(If "Send read requests at startup" is enabled)*	**0** ... 255 (x 1 s)
Thermostat 1	[Checkbox]
Thermostat 2	[Checkbox]
Thermostat 3	[Checkbox]
Thermostat 4	[Checkbox]
Inputs	[Checkbox]
Internal scenes	[Checkbox]
Door phone *(**VERSO** INDOOR only)*	[Checkbox]
Cameras *(**VERSO** INDOOR only)*	[Checkbox]

The *Project name* field can be used to enter a descriptive project name. This will be shown in the System Settings > Info page.

The *Display orientation* parameter allows selecting the configuration of the user interface (portrait or landscape).

The *Enable date/time objects* parameter determines whether the *Date* (DPT 11.001) and *Time* (DPT 10.001) communication objects will be enabled. When enabled, writing to these objects will update the internal date and time of the device. Conversely, when the date/time is modified through the user interface (Preferences > Date & time), the objects will be updated with the new setting.

If the *Date/time sending period* parameter is set to a non-zero value, date and time will also be sent periodically to the bus.

If *Send read requests at startup* is enabled, the device will send read requests to the bus for all the status objects related to the visualisation. This makes it possible to update the displayed states when the device is initially switched on, or after power cycling. The *Startup requests initial delay* parameter controls the initial delay before the device starts to send these read requests.

The *Thermostat 1...Thermostat 4*, *Inputs*, *Internal scenes*, *Door phone*, and *Cameras* parameters enable the corresponding configuration sections. See

3.2.2 Network settings (**VERSO** INDOOR)

Parameter name	Values
IP address	**192.168.1.250**
Subnet mask	**255.255.255.0**
Gateway	**192.168.1.1**
DNS server 1	**8.8.8.8**
DNS server 2	**0.0.0.0**

This section allows configuration of TCP/IP network parameters: IP address, subnet mask, default gateway, and DNS servers.

> **Note**
>
> Make sure to select an IP address that is outside the range of addresses managed by the home router. Otherwise, IP conflicts could arise, resulting in intermittent connectivity problems.

3.2.3 Internal temperature sensor

Parameter name	Values
Calibration offset	-50 ... **0** ... 50 (x 0.1ºC)
Sending period	**0** ... 255 (x 10 s, 0 = Disabled)
Send when change is greater than	**0** ... 255 (x 0.1ºC, 0 = Disabled)

This section allows configuration of the internal temperature probe.

The *Calibration offset* parameter defines the initial value of the calibration offset to be added to the measured temperature values. This offset can also be adjusted from the user interface (System settings > Temperature Probe)

The corrected temperature values are made available through the *Internal temperature* communication object (DPT 9.001). The *Sending period* and *Send when change is greater than* parameters can be used to control when this object is updated (and its value sent to the bus)

3.2.4 Timer groups

Parameter name	Values
Timer group 1 lock object	[Checkbox]
Timer group 2 lock object	[Checkbox]
Timer group 3 lock object	[Checkbox]
Timer group 4 lock object	[Checkbox]

As described in 2.2 Time schedules on page 22, timers can optionally be added to one of four available timer groups. This allows to temporarily lock (disable) groups of timers with a single operation.

The parameters in this section control whether the corresponding lock objects (*Timer group 1 lock … Timer group 4 lock*) are available.

3.3 User Interface

3.3.1 General

Parameter name	Values
Page 1	[Checkbox]
...	
Page 12	[Checkbox]
Favorites	[Checkbox]
Default language	**English** / Deutsch / Español / Français
Default theme	**Light** / Dark
Show time in title bar	No / **Yes**
Show temperature in title bar	No / **Yes**
Show date/time in home page	No / **Yes**
Show temperature in home page	No / **Yes**
Source for temperature display	**Internal sensor**
(Only if temperature is displayed in title bar	Temperature Probe IN1
and/or home page)	Temperature Probe IN2
	Temperature Probe IN3
	Temperature Probe IN4
	External object
Default page after home page	**Menu** / Page 1 ... Page 12 / Favorites
Regulation and rotary controls behaviour	**Send when dialog is confirmed**
	Send on touch release

Checkboxes *Page 1 ... Page 12* and *Favorites* enable the corresponding pages.

Parameter *Default language* selects the default language for system texts. This setting can be modified from the user interface at any time (Preferences > Language).

Parameter *Default theme* selects the default appearance of the user interface (light or dark mode). This setting can be modified from the user interface at any time (Preferences > Theme).

Parameters *Show time in title bar* and *Show temperature in title bar* control what information will be shown in the title bar, besides the name of the current page.

Parameters *Show date/time in home page* and *Show temperature in home page* control what information will be shown in the Home page.

If the temperature is set to be displayed in the title bar and/or in the Home page, an additional parameter *Source for temperature display* allows selecting the source for the displayed temperature: The internal temperature probe, an external probe connected to one of the multifunction inputs, or an external communication object. If the latter is selected, communication object *External temperature* (DPT 9.001) will be enabled.

> **Note**
>
> When using an external probe connected to one of the multifunction inputs as a temperature source, make sure to configure the input accordingly.

Parameter *Default page after home page* controls which page will be the first one shown immediately after the Home page. This can be the Menu page, any of the configured control pages, or the Favorites page.

Parameter *Regulation and rotary controls behaviour* determines the user interface behaviour for regulation bars and rotary controls. The default is to send the final value to the bus just once, when the dialog is confirmed (closed). However, it is also possible to send updated values to the bus while adjusting, every time the touch is released.

3.3.2 Power saving mode

Parameter name	Values
Timeout for home page	0 ... **5** ... 60 (x 1 min, 0 = Disabled)
Timeout for power saving mode	0 ... **5** ... 60 (x 1 min, 0 = Disabled)
Page to show when leaving power saving mode	**Home page** / Default page
Enable wake up object	[Checkbox]

This section controls the behaviour of the power saving mode (see 2.1.9 Power saving mode on page 21). The *Timeout for home page* and *Timeout for power saving mode* parameters define the initial timeout values. Both of them can be adjusted from the user interface (Preferences > Power saving).

Parameter *Page to show when leaving power saving mode* determines which page will be shown when leaving power saving mode. This can be either the Home page or the "default" page defined in the previous section.

A special *Wake up* communication object can optionally be enabled with the *Enable wake up object* parameter. Writing the value "1" to this object wakes the device from power saving mode.

3.3.3 Access control

Parameter name	Values
Restricted access levels	**One level** / Two levels
Additional parameters for "One level":	
Default password:	
- Digit 1	0 … **1** … 9
- Digit 2	0 … **1** … 9
- Digit 3	0 … **1** … 9
- Digit 4	0 … **1** … 9
Additional parameters for "Two levels":	
Default User password:	
- Digit 1	0 … **1** … 9
- Digit 2	0 … **1** … 9
- Digit 3	0 … **1** … 9
- Digit 4	0 … **1** … 9
Default Master password:	
- Digit 1	0 … **2** … 9
- Digit 2	0 … **2** … 9
- Digit 3	0 … **2** … 9
- Digit 4	0 … **2** … 9

Pages in the user interface can be password protected with 4-digit PIN codes (see 2.1.8 Access control on page 20). The parameter *Restricted access levels* defines whether one single password is used for all protected pages ("One level"), or whether separate user and master passwords are used ("Two levels").

The setting of this parameter also determines the available options for configuration parameters that relate to access control for individual pages. These are labeled as [Access control options] in parameter tables. The available options in each case are:

- If "One level" is selected: **No** / Yes

- If "Two levels" is selected: **No** / User password / Master password

Default (initial) passwords are also defined in this section. Passwords can be modified at any time through the user interface (Preferences > Passwords).

3.3.4 Main menu

Parameter name	Values
Menu page layout	**Fixed positions**
	Flexible – center
	Flexible – center + stretch horizontally
	Flexible – center + stretch vertically
	Flexible – center + stretch in both directions
Restrict access to Menu page	[Access control options]
Restrict access to Alarms page	[Access control options]
Restrict access to Timers page	[Access control options]

This section allows configuration of the Menu page layout, and access control for the Menu page, as well as for the Alarms and Timers page.

3.3.5 Settings pages

Parameter name	Values
Restrict access to Settings page	[Access control options]
Restrict access to User Preferences page	[Access control options]
Restrict access to System settings page	[Access control options]

This section allows configuration of access control for settings pages.

3.3.6 Favorites

Parameter name	Values
Favorites page layout	**Fixed positions**
	Flexible – center
	Flexible – center + stretch horizontally
	Flexible – center + stretch vertically
	Flexible – center + stretch in both directions
Restrict access to Favorites page	[Access control options]
Restrict access to Edit Favorites page	[Access control options]

This section allows configuration of the Favorites page layout, and access control for the Favorites and Edit Favorites page.

3.3.7 Touch gestures

Parameter name	Values
Gesture: Up	**Disabled** / Send binary value / Run scene
Value to send *(Only for "Send binary value")*	0 / **1**
Scene type *(Only for "Run scene")*	**External** / Internal
Scene number *(Only for "Run scene")*	**1** ... 64
Gesture: Down	*Same parameters as for Gesture: Up*
Gesture: Left	*Same parameters as for Gesture: Up*
Gesture: Right	*Same parameters as for Gesture: Up*
Gesture: Multitouch	*Same parameters as for Gesture: Up*

Configuration of touch gestures is carried out in this section (see 2.1.10 Touch gestures on page 21). A different action can be configured for each of the five supported touch gestures. This can be either sending a binary value to the bus ("0" or "1"), or triggering a scene.

If a binary action is configured, a communication object (*Gesture: Up*, *Gesture: Down*, *Gesture: Left*, *Gesture: Right*, or *Gesture: Multitouch*) will be enabled. The selected value (*Value to send*) will be written to this communication object.

If a scene action is configured, this can be either an external scene or an internal scene. For external scenes, a scene recall request will be sent through the *Run/save external scenes* object. For internal scenes, the selected scene will be run directly.

> **Note**
>
> If an internal scene shall be run, make sure to enable and configure the "Internal scenes" function (see 3.2.1 General on page 47 and 3.7 Scenes on page 101).

3.4 Control Pages

3.4.1 General

Parameter name	Values
Name	[Text field]
Icon	[Icon selection]
Page layout	**Fixed positions**
	Flexible – center
	Flexible – center + stretch horizontally
	Flexible – center + stretch vertically
	Flexible – center + stretch in both directions
Restrict page access	[Access control options]
Component 1	[Checkbox]
...	
Component 8	[Checkbox]

Each page has a *Name* and optionally an *Icon*. The name and icon are shown in the Menu page. Additionally, the name is shown in the title bar when the page is displayed.

The *Page layout* can also be selected, along with the access control setting (*Restrict page access*).

Additionally, a set of checkboxes are provided to enable or disable each of the eight components that can be used in each page.

3.4.2 Components

Parameter name	Values
Name	[Text field]
Component type	**Indicator**
	Indicator with Alarm
	Push Button
	Double Push Button
	Regulation Bar
	Rotary Control
	Numeric Keypad
	Dimmer Control
	Shutter Control
	RGBW Control
	Temperature Control
	Door phone page link
	Cameras page link

All components have a *Name*, which is displayed in the visualisation.

A *Component type* must also be selected. Available configuration options are specific to each component type, and are described separately in the following sections.

The "Door phone page link" and "Cameras page link" component types are only available for **VERSO** INDOOR.

Note that there is no need for a dedicated "Timer" component type, as most component types support a timer function (see 3.4.2.13 Timer function on page 87).

3.4.2.1 Indicator

Parameter name	Values
Indicator type	**Binary**
	Enumerated
	Numerical
	Level bar
	Level dial

Indicators can be used for the visualisation of status values. They display the current value of an associated communication object.

Several indicator types are supported. Each type is described separately.

Binary

Parameter name	Values
Text for 0/Off	[Text field]
Icon for 0/Off	[Icon selection]
- Icon colour	[Icon colour]
Text for 1/On	[Text field]v
Icon for 1/On	[Icon selection]
- Icon colour	[Icon colour]

Binary indicators display the state of an associated binary (DPT 1) object, *Binary indicator*.

Both an icon and/or a text label can be associated to each of the two possible states of the communication object ("1"/On and "0"/Off).

Enumerated

Parameter name	Values
Number of states	**1** … 6
Value for state 1	**0** … 255
Text for state 1	[Text field]
Icon for state 1	[Icon selection]
- Icon colour	[Icon colour]
States 2 … 6	*Same parameters as for state 1*
Define fallback state	[Checkbox]
Text for fallback state *(If fallback state is enabled)*	[Text field]
Icon for fallback state *(If fallback state is enabled)*	[Icon selection]
- Icon colour	[Icon colour]

Enumerated indicators can display several different states, depending on the value of an associated 1-byte (DPT 5.010) communication object. Values can range from 0 to 255.

Up to 6 states can be defined. Each of these states is associated with a specific value of the communication object. For each state, it is possible to define both an icon and/or a text label. The icon and/or text will be shown when the corresponding value is received through the communication object.

Additionally, it is possible to define the icon and/or text label for a so called "fallback state". The fallback state icon and/or text will be shown when a value that is not associated to any other state is received through the communication object.

If no fallback state is defined, received values that do not match any state will be ignored.

Numerical

Parameter name	Values
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
	4-byte unsigned int
	4-byte signed int
	4-byte float
Decimal places *(Only for float data types)*	**Automatic** / 0 / 1 / 2
Units *(Except for "Percentage" data type)*	[Text field]

Numerical indicators display the value of the associated communication object as a number. The following table shows the available data types for the communication object and the corresponding value ranges:

Data type	Value range
1-byte unsigned int	0 ... 255
1-byte signed int	-128 ... 127
Percentage	0% ... 100%
2-byte unsigned int	0 ... 65535
2-byte signed int	-32768 ... 32767
2-byte float	-671088.64 ... 670760.96
4-byte unsigned int	0 ... 4294967295
4-byte signed int	-2147483648 ... 2147483647
4-byte float	-3.4028E+38 ... 3.4028E+38

For the 2-byte and 4-byte float data types, it is possible to define the number of decimal places to show (*Decimal places* parameter). This can be set to 0, 1, 2, or Automatic; with the "Automatic" option, the number of decimal places to show is determined automatically by the device, depending the actual object value:

- Two decimal places if the absolute value is less than 1

- One decimal place if the absolute value is 1 or greater, and less than 100
- No decimal places if the absolute value is 100 or greater

Additionally, for the 4-byte float data type, large values (absolute value greater than 1000000) are always represented using scientific notation.

Finally, for all data types except "Percentage", a text suffix (*Units* parameter) can be added to represent value units, such as "°C". This suffix will be displayed immediately after the numeric object value.

Level bar

Parameter name	Values
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
Decimal places *(Only for "2-byte float" data type)*	**Automatic** / 0 / 1 / 2
Units *(Except for "Percentage" data type)*	[Text field]
Minimum bar value	[See description]
Maximum bar value	[See description]

Level bar indicators are similar to numerical indicators, and support the same data types (except for 4-byte data types), but the value of the associated communication object is represented as a level bar. In addition, the value is also displayed in textual form above the level bar.

The *Minimum bar value* and *Maximum bar value* parameters allow to limit the effective range of values that are used to determine the bar level. Values that are out of this range will be represented as an empty or full bar, respectively. The textual representation of the value is not affected by these parameters.

> **Example**
>
> A level bar with the data type set to 2-byte float can be used to represent temperature values. The "minimum bar value" and "maximum bar value" parameters can be used to

> define the range of temperatures to be represented in the level bar; for example, 0 to 60ºC. If a value of 80ºC is received, the level bar would be represented as full (as for any value above 60ºC), however the actual numeric value (80ºC) would still be displayed above the bar.

The range of valid values for the *Minimum bar value* and *Maximum bar value* parameters equals the range of the selected data type. The default value for the parameter will be the minimum or maximum value of the valid range, respectively.

Level dial

Parameter name	Values
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
Decimal places *(Only for "2-byte float" data type)*	**Automatic** / 0 / 1 / 2
Units *(Except for "Percentage" data type)*	[Text field]
Minimum dial value	[See description]
Maximum dial value	[See description]

Level dial indicators are completely equivalent to level bar indicators. The only difference is that for the graphical representation of the value, a circular dial is used instead of a level bar.

3.4.2.2 Indicator with Alarm

Parameter name	Values
Alarm trigger value	0 / **1**
Alarm acknowledge value	0 / **1**

Indicators with Alarm function are regular indicator components that in addition can be used to detect alarm or fault conditions.

For this, two additional communication objects are enabled: *Alarm trigger* and *Alarm acknowledge*. The activation value for both objects is selected via the parameters *Alarm trigger value* and *Alarm acknowledge value*, respectively.

See 2.3 Alarms on page 25 for a detailed description of the alarm function.

3.4.2.3 Push Button

Parameter name	Values
Push Button type	**Binary**
	Choice
	Numerical
	Scene

Push button components allow sending predefined values to the bus through an associated communication object (the "control" object).
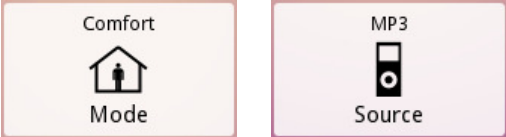
In some cases, depending on the component configuration, an additional communication object is enabled (the "indicator" or status object). When new values are received from the bus through this object, the component status is automatically updated.

Several push button types are supported. Each type is described separately.

Binary

Parameter name	Values
Action	**Send 0**
	Send 1
	Toggle
	Short 0, long 1
	Short 1, long 0
	Press 0, release 1
	Press 1, release 0
Parameters for "Send 0" and "Send1":	
Button text	[Text field]
Button icon	[Icon selection]
- Icon colour	[Icon colour]
Parameters for all other actions:	
Button text for 0/Off	[Text field]
Button icon for 0/Off	[Icon selection]
- Icon colour	[Icon colour]
Button text for 1/On	[Text field]
Button icon for 1/On	[Icon selection]
- Icon colour	[Icon colour]

When binary push buttons are pressed, a value is sent to the bus through the associated binary (DPT 1) communication object, *Binary control*.

The specific behaviour of the button depends on the *Action* parameter. The following choices are available:

- *Send 0:* Sends the value "0" (off) when the button is pressed.

- *Send 1:* Sends the value "1" (on) when the button is pressed.

- *Toggle:* Alternates sending "0" (off) and "1" (on) on each button press (the first time, the value "1" (on) is sent).

- *Short 0, Long 1:* Sends "0" on a short press, "1" on a long press.

- *Short 1, Long 0:* Sends "1" on a short press, "0" on a long press.

- *Press 0, Release 1:* Sends "0" when the user starts pressing the button, and "1" when the button is released.

- *Press 1, Release 0:* Sends "1" when the user starts pressing the button, and "0" when the button is released.

For actions that send a single fixed value (*Send 0* and *Send 1*), it is possible to define both an icon and/or text label that will be shown in the push button component.

For actions that can send two different values (all other actions), it is possible to define separate icons and/or text labels for the values "0" and "1". The component will show the icon and/or text corresponding to the last value that was sent.

Choice

Parameter name	Values
Number of options	**1** ... 6
Option 1 value	**0** ... 255
Option 1 text	[Text field]
Option 1 icon	[Icon selection]
- Icon colour	[Icon colour]
Options 2 ... 6	*Same parameters as for option 1*
Show fixed icon/text	**No** / Yes
Button text *(If fixed icon/text is selected)*	[Text field]
Button icon *(If fixed icon/text is selected)*	[Icon selection]
- Icon colour	[Icon colour]

Choice Push Buttons can be used to send one of several predefined values to the bus through the associated 1-byte (DPT 5.010) communication object. Values can range from 0 to 255.

Up to 6 options can be defined. Each of these options is associated with a value to be sent to the bus. For each option, it is possible to define both an icon and/or a text label. When the button is pressed, a pop-up dialog will be shown allowing the user to select one of the available options. The value associated with the selected option will then be sent to the bus.
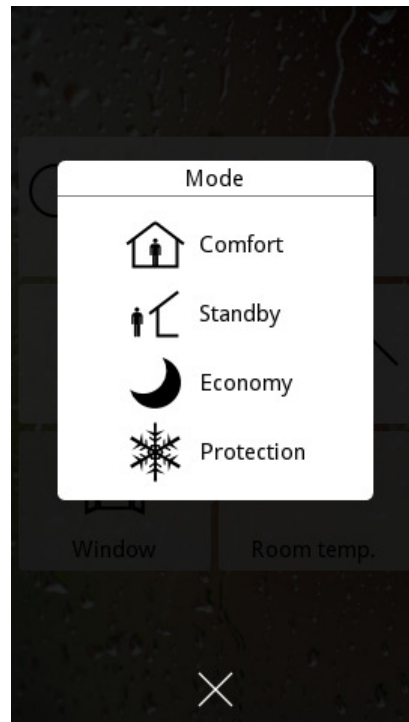
By default, the push button will show the icon and/or text corresponding to the last option that was selected. However, if the *Show fixed icon/text* parameter is set to "Yes", a fixed icon and/or text will be displayed on the component instead.

Numerical

Parameter name	Values
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
Action	**Send fixed value**
	Send A/B values on short/long press
Parameters for "Send fixed value":	
Value	[See description]
Button text	[Text field]
Button icon	[Icon selection]
- Icon colour	[Icon colour]
Parameters for "Send A/B values...":	
Value A	[See description]
Text for value A	[Text field]
Icon for value A	[Icon selection]
- Icon colour	[Icon colour]
Value B	[See description]
Text for value B	[Text field]
Icon for value B	[Icon selection]
- Icon colour	[Icon colour]

Numerical push buttons are similar to binary push buttons, but allow sending either a fixed numeric value, or one of two possible numeric values to the associated communications object.

The specific behaviour of the button depends on the *Action* parameter. The following choices are available:

- *Send fixed value*: Sends a fixed value when the button is pressed.

- *Send values A/B on short/long press*: Sends a configured value (A) on short press, and another value (B) on long press.

The range of valid values for the *Value*, *Value A*, and *Value B* parameters equals the range of the selected data type. The default value for all three parameters is 0.

Scene

Parameter name	Values
Scene type	**External** / Internal
Scene number	**1** ... 64
Allow saving scene	No / **Yes**
Button text	[Text field]
Button icon	[Icon selection]
- Icon colour	[Icon colour]

Scene push buttons allow activating or saving a scene. The scene can be external or internal (*Scene type* parameter). The *Scene number* parameter allows specifying the target scene (1-64).

For external scenes, the scene recall or store request will be sent through the global *Run/save external scenes* object. For internal scenes, the request is not sent to the bus; instead, it is processed internally (the internal scene controller will recall or store the selected scene).

> **Note**
>
> When using internal scenes, make sure to enable and configure the "Internal scenes" function (see 3.2.1 General on page 47 and 3.7 Scenes on page 101).

Further, the *Allow saving scene* parameter controls whether the push button will allow only activation of the associated scene, or whether it will also allow saving. If saving of the scene is allowed, then a short press will trigger activation of the scene, while a long press will trigger saving of the scene. If saving of the scene is not allowed, then pressing the button will always activate the scene.

The *Button text* and *Button icon* parameters can be used to set a text label and/or icon to be displayed on the component.

3.4.2.4 Double Push Button

Parameter name	Values
Double push button type	**Binary**
	Enumerated
	Numerical
	Scene
Left button icon	[Icon selection]
- Icon colour	[Icon colour]
Right button icon	[Icon selection]
- Icon colour	[Icon colour]

Double push buttons combine the function of an Indicator component with an additional pair of buttons.

The indicator displays the value of the associated "indicator" communication object. The left and right buttons are used to select values to be sent to the bus through an additional "control" communication object. Whenever a value is sent to the bus through the "control" object, the indicator display is also updated automatically.
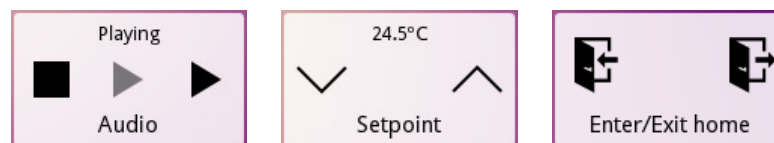
Several types of double push button components are supported. Each type is described separately.

Binary

Parameter name	Values
Action	**Left 0, right 1** / Left 1, right 0
Text for 0/Off	[Text field]
Icon for 0/Off	[Icon selection]
- Icon colour	[Icon colour]
Text for 1/On	[Text field]
Icon for 1/On	[Icon selection]
- Icon colour	[Icon colour]

Double push buttons of type "Binary" are based on Binary Indicators. The indicator function is configured in the same way as for the corresponding indicator component.

When the left or right buttons are pressed, a fixed value will be sent to the bus through the *Binary control* communication object (DPT 1). The specific values sent depend on the setting of the *Action* parameter. The following choices are available:

- *Left 0, Right 1:* Pressing the left button sends the value "0" (off). Pressing the right button sends "1" (on)

- *Left 1, Right 0:* Pressing the left button sends the value "1" (on). Pressing the right button sends "0" (off)

Enumerated

Parameter name	Values
Number of states	**1** ... 6
Value for state 1	**0** ... 255
Text for state 1	[Text field]
Icon for state 1	[Icon selection]
- Icon colour	[Icon colour]
States 2 ... 6	*Same parameters as for state 1*
Define fallback state	[Checkbox]
Text for fallback state *(If fallback state is enabled)*	[Text field]
Icon for fallback state *(If fallback state is enabled)*	[Icon selection]
- Icon colour	[Icon colour]

Double push buttons of type "Enumerated" are based on Enumerated Indicators. The indicator function is configured in the same way as for the corresponding indicator component: Up to 6 options ("states") can be defined, each associated with a specific value, along with an optional "fallback state".

The left and right buttons allow users to select one of available options. Once an option is selected, the associated value is sent to the bus through the "control" communication object (DPT 5.010).

Numerical

Parameter name	Values
Action	**Left decrease, right increase**
	Left increase, right decrease
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
Decimal places *(Only for "2-byte float" data type)*	**Automatic** / 0 / 1 / 2
Units *(Except for "Percentage" data type)*	[Text field]
Short step	[See description]
Long step	[See description]
Minimum value	[See description]
Maximum value	[See description]
Initial value	[See description]

Double push buttons of type "Numerical" are based on Numerical Indicators. The indicator function is configured in the same way as for the corresponding indicator component.

The left and right buttons allow the user to select a value from a given (configurable) range. By default, the left button is used to decrease the value, and the right button to increase it; however the opposite behaviour can also be selected through the *Action* parameter.

Minimum and maximum values for the allowed range must be defined (*Minimum value* and *Maximum value* parameters). The component will not allow the user to select a value outside this range. Also, the *Short step* and *Long step* values must be configured. The short step determines the increment or decrement that is applied to the current value on a short press of any of the two buttons. The long step determines the increment or decrement that is applied (repeatedly) to the current value when any of the two buttons is kept pressed, and until it is released.

Once a value is selected, it will be sent to the bus through the "control" communication object.

The valid range for the *Short step*, *Long step*, *Minimum value*, *Maximum value*, and *Initial value* parameters equal the range of the selected data type. The default values for each parameter is listed next:

- Short step: 1
- Long step: 5
- Minimum value: Minimum valid value for the selected data type
- Maximum value: Maximum valid value for the selected data type
- Initial value: 0

Scene

Parameter name	Values
Left: Scene type	**External** / Internal
Left: Scene number	**1** … 64
Left: Allow saving	No / **Yes**
Right: Scene type	**External** / Internal
Right: Scene number	**1** … 64
Right: Allow saving	No / **Yes**

Double push buttons of type "Scene" behave exactly in the same way as Scene Push Button components, except that each of the two buttons (left and right) can be individually configured for a different scene.

3.4.2.5 Regulation Bar

Parameter name	Values
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
Decimal places *(Only for "2-byte float" data type)*	**Automatic** / 0 / 1 / 2
Units *(Except for "Percentage" data type)*	[Text field]
Minimum bar value	[See description]
Maximum bar value	[See description]

Regulation Bars are based on Indicators of type "Level bar". They display the current value of the associated communication object as a level bar, and their configuration parameters are exactly the same as for the corresponding indicator component.
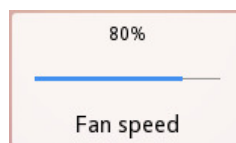
Additionally, when the component area is pressed, a window will pop up, containing an interactive regulation bar that allows the user to select values to be sent to the bus.
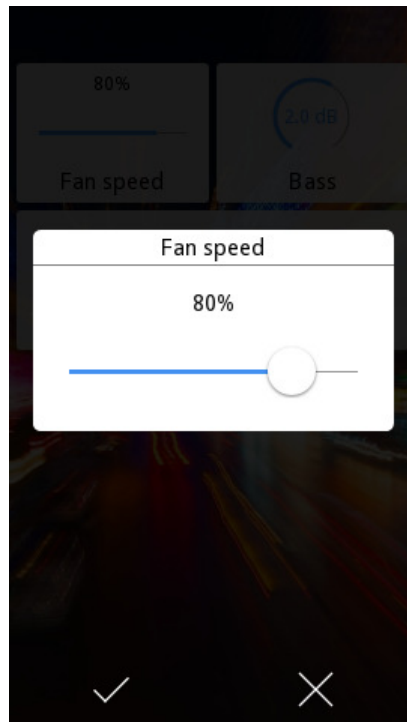
Values are sent to the bus through an additional "control" communication object. Whenever a value is sent to the bus through the "control" object, the indicator display is also updated automatically.

> **Note**
>
> The interactive regulation bar only allows the selection of values within the range configured through the *Minimum bar value* and *Maximum bar value* parameters.

3.4.2.6 Rotary Control

Parameter name	Values
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
Decimal places *(Only for "2-byte float" data type)*	**Automatic** / 0 / 1 / 2
Units *(Except for "Percentage" data type)*	[Text field]
Minimum dial value	[See description]
Maximum dial value	[See description]

Rotary Controls are based on Indicators of type "Level dial". They display the current value of the associated communication object as a level dial, and their configuration parameters are exactly the same as for the corresponding indicator component.
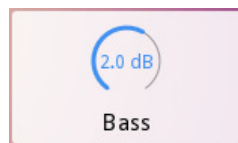
Additionally, when the component area is pressed, a window will pop up, containing an interactive rotary control that allows the user to select values to be sent to the bus.
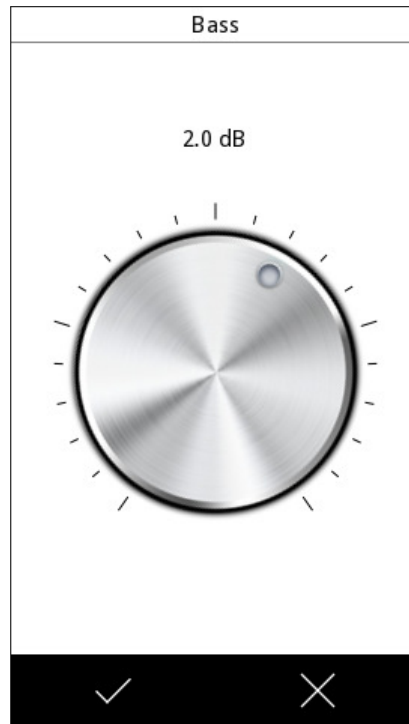
Values are sent to the bus through an additional "control" communication object. Whenever a value is sent to the bus through the "control" object, the indicator display is also updated automatically.

> **Note**
>
> The interactive rotary control only allows the selection of values within the range configured through the *Minimum dial value* and *Maximum dial value* parameters.

3.4.2.7 Numeric Keypad

Parameter name	Values
Data type	**1-byte unsigned int**
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
Decimal places *(Only for "2-byte float" data type)*	**Automatic** / 0 / 1 / 2
Units *(Except for "Percentage" data type)*	[Text field]
Minimum input value	[See description]
Maximum input value	[See description]
Error text for invalid input	[Text field]

Numeric Keypads are based on Indicators of type "Numerical". They display the current value of the associated communication object as a number, and can be configured in a similar way as the corresponding indicator component.
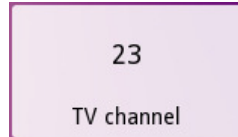
Additionally, when the component area is pressed, a window will pop up, containing an interactive numeric keypad that allows the user to enter values to be sent to the bus.
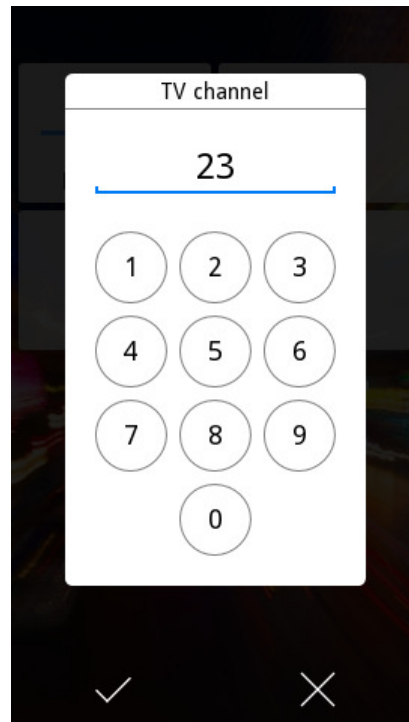
Values are sent to the bus through an additional "control" communication object. Whenever a value is sent to the bus through the "control" object, the indicator display is also updated automatically.

Three additional parameters are defined: The *Minimum input value* and *Maximum input value* parameters limit the range of valid values that can be entered through the numeric keypad. If users enter a value that is outside this range, the value will not be accepted; instead, an error message will be shown. The error text can be customized through the *Error text for invalid input* parameter.

The range of valid values for the *Minimum input value* and *Maximum input value* parameters equals the range of the selected data type. The default value for the parameter will be the minimum or maximum value of the valid range, respectively.

3.4.2.8 Dimmer Control

Parameter name	Values
Dimmer type	**Relative dimmer** / Absolute dimmer
Action	**Left off, right on** / Left on, right off
Show status value *(Only for relative dimmers)*	[Checkbox]
Dimming step *(Only for relative dimmers)*	**100%** / 50% / 25% / 12.5% / 6.25% / 3.1% / 1.6%
Left button icon	[Icon selection]
- Icon colour	[Icon colour]
Right button icon	[Icon selection]
- Icon colour	[Icon colour]

Dimmer Controls are specialized components that can be used to control light dimmer actuators.

The following communication objects are available:

- *Dimming level indicator* (DPT 5.001)
- *Light on/off* (DPT 1.001)
- One of:
    - *Relative dimming control* (DPT 3.007)
    - *Absolute dimming control* (DPT 5.001)

Dimmer controls are represented as double push buttons in the visualisation.
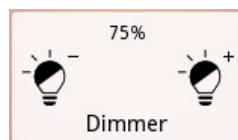
The current light intensity value is displayed as a percentage (0…100%), and is updated whenever a new value is received through the *Dimming level indicator* communication object.

Some relative dimmers do not provide feedback on the current light intensity. In this case, the status value can be disabled via the *Show status value* parameter.

The left and right buttons are used to switch the light on and off, and to adjust light intensity. The *Action* parameter allows selecting the function of each button (*Left off, right on*, or *Left on, right off*).

A short press on any of the buttons sends an On / Off command to the bus through the *Light on/off* communication object.

Handling of long presses depend on the type of dimmer being controlled (as defined by the *Dimmer type* parameter).

Relative dimmers

A long press on any of the buttons starts an increase / decrease light intensity operation. A dimming command will be sent to the dimmer actuator through the *Relative dimming control* communication object, requesting that the light intensity is increased (or decreased) by the amount defined by the *Dimming step* parameter. When the button is released, a "stop" command will be sent to the dimmer; if the dimming operation was still in progress, it will be interrupted.

> **Note**
>
> It is common to set the *Dimming step* to 100%. This way, a long press on any of the buttons will initiate a dimming operation which spans the entire dimming range, all the way up to 100% or down to 0%. As soon as the light intensity reaches the desired value, the user can release the button and the dimming operation will be interrupted.

Absolute dimmers

A long press on any of the buttons will bring up a window with a regulation bar. The regulation bar allows precise selection of light intensity values (0...100%). The selected value will be sent to the bus through the *Absolute dimming control* communication object.
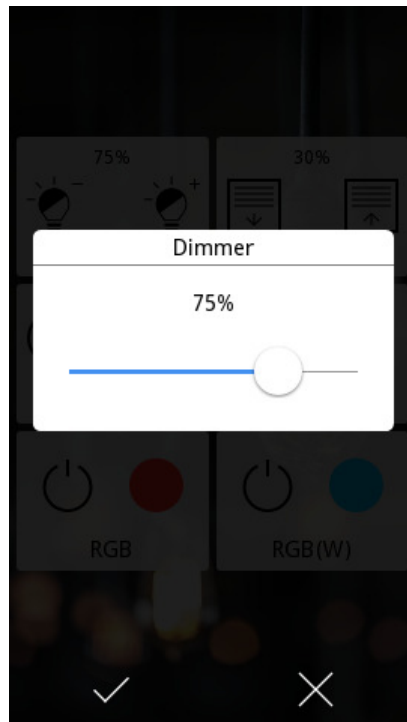
3.4.2.9 Shutter Control

Parameter name	Values
Shutter actuator type	**Basic** / With absolute positioning
Action	**Left down, right up** / Left up, right down
Show status value *(Only for basic shutters)*	[Checkbox]
Left button icon	[Icon selection]
- Icon colour	[Icon colour]
Right button icon	[Icon selection]
- Icon colour	[Icon colour]

Shutter controls are specialized components that can be used to control shutter or blind actuators.

The following communication objects are available:

- *Shutter position indicator* (DPT 5.001)
- *Shutter step/stop* (DPT 1.007)
- One of:

- *Shutter up/down* (DPT 1.008)
- *Shutter position control* (DPT 5.001)

Shutter controls are represented as double push buttons in the visualisation.
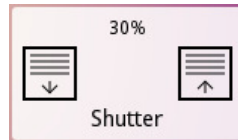
The current shutter / blind position is displayed as a percentage (0...100%), with 0% corresponding to the "completely open" position, and 100% to the "completely closed" position. This is updated whenever a new value is received through the *Shutter position indicator* communication object.

Some basic shutter actuators do not provide feedback on the current shutter / blind position. In this case, the status value can be disabled via the *Show status value* parameter.

The left and right buttons are used to control the shutter actuator. The *Action* parameter allows selecting the function of each button (*Left down, right up*, or *Left up, right down*).

A short press on any of the buttons sends a step up or step down command to the bus through the *Shutter step/stop* communication object. If a move operation was in progress, it will be interrupted. Otherwise, a single step operation is performed.

Handling of long presses depend on the type of shutter actuator being controlled (as defined by the *Shutter actuator type* parameter).

Basic shutter actuators

A long press on any of the buttons starts a move up or move down operation through the *Shutter up/down* communication object. This operation will ultimately open or close the shutter completely, unless it is interrupted.

Shutter actuators with absolute positioning

A long press on any of the buttons will bring up a window with a regulation bar. The regulation bar allows precise selection of shutter position values (0...100%). The selected value will be sent to the bus through the *Shutter position control* communication object.
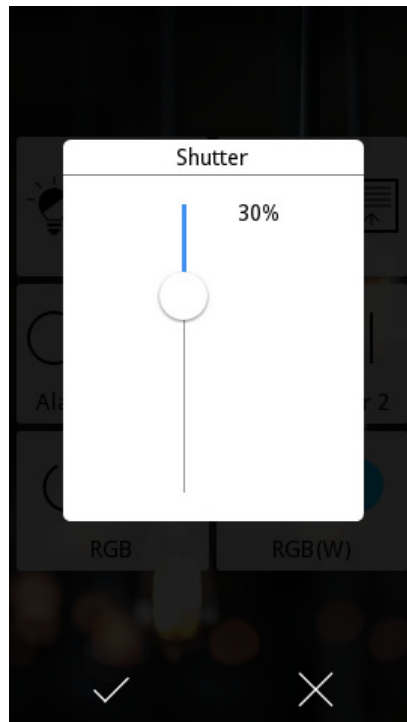
3.4.2.10 RGBW Control

Parameter name	Values
Use white channel	**No** / Yes
RGB object type	Three separate R,G,B objects (DPT 5.001)
	One combined RGB object (DPT 232.600)
Left button function	**Switch off** / Toggle
Left button icon	[Icon selection]
- Icon colour	[Icon colour]

RGBW Controls are specialized components that can be used to control both RGB and RGBW (RGB + White) LED dimmer actuators.

The *RGB object type* parameter sets whether the RGBW control should use three separate 1-byte R, G, B communication objects (one per channel), or one combined 3-byte RGB object.

The *Use white channel* parameter selects whether a white channel should also be controlled, in addition to the Red, Green, and Blue channels. The white channel is controlled through an additional 1-byte object.

Operation

The RGBW Control shows two buttons; an action button (left) and the colour selection button (right):

The function of the action button is configured through the *Left button function* parameter: Either switch all channels off (sets all channels to 0%), or toggle between switching all channels off and recalling the previously selected colour.

The colour selection button shows the current colour. Pressing this button brings up an interactive colour selection window. This allows selecting different colours and sending them to the bus. The user can also save their "favorite" colours for later use. For each RGBW Control, up to ten favorite colours can be saved.
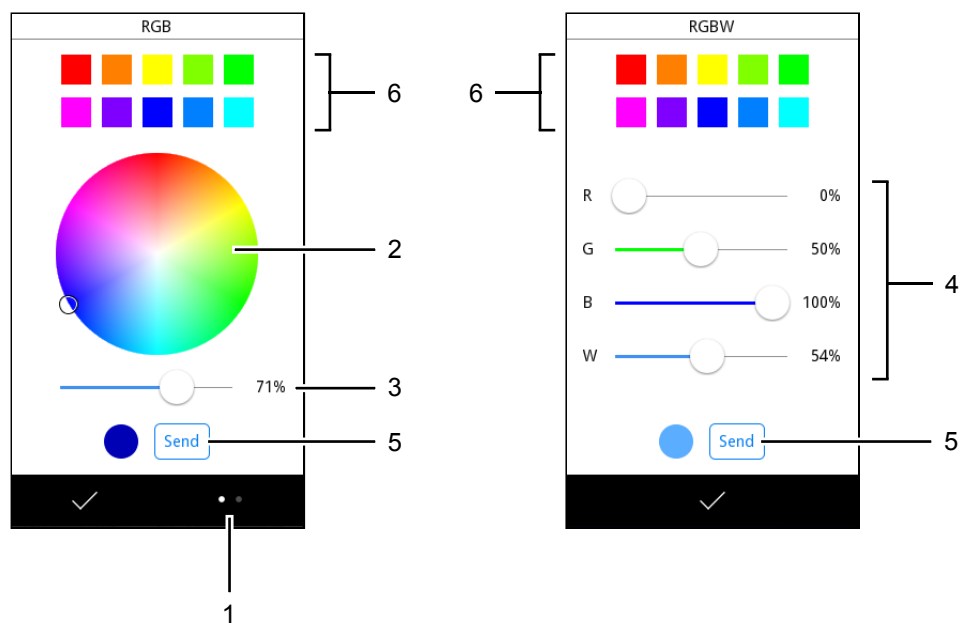
While the colour selection window is open:

- Use the "view switch" button (1) to toggle between the "colour wheel" and "colour bars" views. This is only available if white channel control is not enabled; if the component is configured to control a white channel, the colour bars view is always used.

- Colour wheel: Use the wheel (2) to select a base colour; then use the bar (3) to select the level of brightness.

- Colour bars: Use the bars (4) to select the desired value for each channel.

- Press the *Send* button (5) to send the selected colour to the bus.

- Perform a long press on any of the "favorite colour" positions (6) in order to save the currently selected colour in that position.

- Perform a short press on one of the "favorite colour" positions (6) in order to recall the stored colour.

3.4.2.11 Temperature Control

Parameter name	Values
Temperature control type	**Absolute**
	Relative (offset control)
	Relative (binary stepping)
Left button icon	[Icon selection]
- Icon colour	[Icon colour]
Right button icon	[Icon selection]
- Icon colour	[Icon colour]

Temperature controls are specialized components that can be used to adjust a temperature value, for example the setpoint in a thermostat.

Temperature controls are represented as double push buttons in the visualisation. The component displays the current temperature from the *Temperature indicator* communication object. The left and right buttons allow selecting new values to be sent to the bus. The actual operation details depend on the *Temperature control type* parameter; each type is described separately.
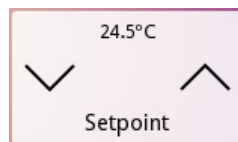
Absolute

Parameter name	Values
Action	**Left decrease, right increase**
	Left increase, right decrease
Short step	0 … **5** … 255 (x 0.1ºC)
Long step	0 … **10** … 255 (x 0.1ºC)
Minimum value	-30 … **15** … 100 (ºC)
Maximum value	-30 … **40** … 100 (ºC)
Initial value	-30 … **22** … 100 (ºC)

Communication objects:

- *Temperature indicator* (2-byte float)
- *Temperature control* (2-byte float)

The left and right buttons allow selection of an absolute temperature value that will be sent to the bus. It is possible to configure the minimum and maximum temperature values that can be selected, as well as the increment or decrement that is applied on a short or long press of the buttons.

Whenever a temperature value is selected, it will be written to the *Temperature control* object.

Relative (Offset Control)

Parameter name	Values
Action	**Left decrease, right increase**
	Left increase, right decrease
Short step	0 … **5** … 255 (x 0.1ºC)
Long step	0 … **10** … 255 (x 0.1ºC)
Minimum offset	-20 … **-10** … 20 (ºC)
Maximum offset	-20 … **10** … 20 (ºC)
Initial base value	-30 … **22** … 100 (ºC)

Communication objects:

- *Temperature indicator* (2-byte float)
- *Temperature offset control* (2-byte float)
- *Temperature offset status* (2-byte float)

In this case, the actual temperature value is assumed to be the sum of two components: A **base** temperature, and a temperature **offset**:

Absolute temperature = Base temperature + temperature offset

This type of temperature control allows adjusting the offset value. Offset values are sent to the bus through the *Temperature offset control* object (2-byte float). However, the displayed temperature value is always the absolute temperature value. Thus, as far as the end user is concerned, the user interface is identical to the Absolute temperature control case: The user simply selects a new temperature value and need not be aware of the fact that the control is internally updating the offset value only.

It is possible to configure the minimum and maximum offset values that can be sent, as well as the increment or decrement that is applied on a short or long press of the buttons.

When new values are received from the bus through the *Temperature indicator* object, the internal temperature offset value remains unchanged, and the internal base temperature value is updated so that the resulting absolute temperature matches the received value.

Additionally, a *Temperature offset status* object is available in order to externally update the internal temperature offset value.

Relative (Binary Stepping)

Parameter name	Values
Action	**Left decrease, right increase**
	Left increase, right decrease
Step	0 … **5** … 255 (x 0.1ºC)
Initial value	-30 … **22** … 100 (ºC)

Communication objects:

- *Temperature indicator* (2-byte float)
- *Temperature step control* (binary)

On each press of the left or right button, a "decrease" (0) or "increase" (1) command is transmitted to the bus through the *Temperature step control* object. The displayed temperature value will be decremented or incremented by the amount defined through the *Step* parameter, thus providing immediate feedback in the visualisation.

3.4.2.12 Page link (**VERSO** INDOOR)

Parameter name	Values
Button icon	[Icon selection]

"Page link" components are used to link from control pages to the Door phone and Cameras pages.

These components are only supported for **VERSO** INDOOR.

3.4.2.13 Timer function

Parameter name	Values
Enable timer function	[Checkbox]
Add to timer group *(Only if timer function is enabled)*	**None** / Group 1 / Group 2 / Group 3 / Group 4

For most components in the visualisation, a timer function can be enabled. When the timer function is enabled for a component, users can setup actions to be run at scheduled times.

The *Enable timer function* checkbox enables the timer function for the component. The *Add to timer group* parameter allows adding a component to one of the four available timer groups. See 2.2 Time schedules on page 22 for a detailed functional description of timers and timer groups.

> **Note**
>
> If timer groups shall be used, make sure to enable the corresponding communication objects. See 3.2.4 Timer groups on page 49.

Timer function support

Component type	Timer function support	Additional notes
Indicator	Not applicable	
Indicator with Alarm	Not applicable	
Push Buttons	Yes	See note 1
Double Push Buttons	Yes	
Regulation Bars	Yes	
Rotary Controls	Yes	
Numeric Keypads	Yes	
Dimmer Controls	Yes	See note 2
Shutter Controls	Yes	See note 3
RGBW Controls	Yes	
Temperature Controls	Yes	See note 4
Door phone page link	Not applicable	
Cameras page link	Not applicable	

Notes:

1. The timer function is supported for the following subtypes:

   - Binary: Send 0 / Send 1 / Toggle
   - Numeric: Send fixed value
   - Choice
   - Scene

2. For absolute Dimmer Controls, the timer function allows scheduling of any dimming action. For relative Dimmer Controls, only on / off actions can be scheduled.

3. For Shutter Controls with absolute positioning, the timer function allows scheduling of any positioning action. For basic Shutter Controls, only simple move up / move down actions can be scheduled.

4. The timer function is supported for absolute Temperature Controls.

3.5 Thermostats

The *Thermostat 1...Thermostat 4* configuration sections will be available if the corresponding functions have been enabled in the Main configuration section.

Thermostats 1 and 2 support two configuration modes: "Complete" (more flexible, but also more complex) and "Simplified". Thermostats 3 and 4 only support the simplified configuration mode.

The remainder of this section describes the parameters corresponding to the complete configuration mode. The simplified configuration mode is described in detail in 3.5.7 Simplified configuration mode on page 96.

For a detailed functional description, see 2.4 Thermostats on page 29.

3.5.1 General

Thermostat function

Parameter name	Values
Configuration	**Complete** / Simplified
Thermostat function	**Heating only** / Cooling only / Heating and cooling
Heating / Cooling switching *(Only if "Heating and cooling" is selected)*	**Manual** / Automatic
Changeover protection band *(Only for Automatic switching of Heating / Cooling)*	**5** ... 100 (x 0.1°C)
Heating / Cooling mode after programming *(Only if "Heating and cooling" is selected)*	**Heating** / Cooling

The *Configuration* parameter selects the configuration mode (complete or simplified). This parameter is only available for thermostats 1 and 2.

The *Thermostat function* parameter allows selecting the thermostat function: Heating only, cooling only, or both heating and cooling. Depending on the selected option, additional "Heating" and "Cooling" configuration sections will be enabled.

If "Heating and cooling" is selected, the *Heating / Cooling switching* parameter determines whether switching between heating and cooling will be done manually or automatically.

- Manual switching: Switching is done through a *Heating / Cooling mode* communication object. Writing a "1" to this object switches to Heating mode; writing a "0" switches to Cooling mode.

- Automating switching: The thermostat automatically manages the changeover between heating and cooling. In this case, the *Changeover protection band* parameter is also enabled. See 2.4.4 Heating and Cooling on page 33 for a detailed description of the automatic switching logic.

In both cases, the current thermostat function (heating or cooling) is available at any time through the *Heating/Cooling mode (status)* communication object.

Also, the *Heating / Cooling mode after programming* parameter defines the initial mode (either "Heating" or "Cooling") after an ETS download.

Thermostat on/off control

Parameter name	Values
Thermostat always ON	**No** / Yes
State after bus recovery *(Only if thermostat is NOT always ON)*	**Last state** / Off / On
Switch ON upon operating mode reception *(Only if thermostat is NOT always ON)*	**No** / Yes

The *Thermostat always ON* parameter determines whether the thermostat should always be on, or whether it should be possible to switch it on and off through an associated communication object.

If the thermostat is NOT always ON:

- Two communication objects are enabled (*On/Off* and *On/Off (status)*) to control operation of the thermostat.

- A parameter *State after bus recovery* is available to define the state of the thermostat at startup, or after a bus recovery event (On, Off, or the last known state)

- A parameter *Switch ON upon operating mode reception* determines if the thermostat should automatically switch on when it receives a request to change the current operating mode, or when it enters "Forced protection" mode via activation of the *Window status* object (if enabled).

Sending of status values

Parameter name	Values
Status sending delay after bus recovery	**0** ... 255 (x 1s, 0 = Disabled)

Parameter *Status sending delay after bus recovery* controls whether the initial value of the thermostat's status objects is sent to the bus at startup or after a bus recovery event. Values can be sent after a configurable delay (1-255 seconds) or this function can be disabled by setting the parameter to 0.

3.5.2 Room Temperature

Parameter name	Values
Room temperature	**Temperature Source 1**
	75% Temp. Source 1 - 25% Temp. Source 2
	50% Temp. Source 1 - 50% Temp. Source 2
	25% Temp. Source 1 - 75% Temp. Source 2
Temperature Source 1	**Internal sensor**
	Temperature Probe IN1
	Temperature Probe IN2
	Temperature Probe IN3
	Temperature Probe IN4
	External object
Temperature Source 2	Internal sensor
(If enabled)	**Temperature Probe IN1**
	Temperature Probe IN2
	Temperature Probe IN3
	Temperature Probe IN4

These parameters allow selection of the temperature sources used to measure room temperature. If more than one source is selected, their values can be averaged in a configurable proportion (75% / 25%, 50% / 50%, or 25% / 75%).

For each temperature source it is possible to select the internal temperature sensor, an external temperature probe connected to one of the four available multifunction inputs (IN1...IN4), or (for temperature source 1), an external communication object. If the latter is selected, a communication object *External temperature source* (DPT 9.001) will be enabled for this purpose.

> **Note**
>
> When using an external probe connected to one of the multifunction inputs as a temperature source, make sure to configure the input accordingly.

3.5.3 Setpoints

Parameter name	Values
Setpoint method	**Absolute** / Relative

The *Setpoint method* parameter determines which method will be used to define setpoints for the thermostat operating modes: Either the Absolute Setpoints method, or the Relative Setpoints method.

See 2.4.3 Setpoints on page 29 for a detailed description of operating modes, absolute setpoints, and relative setpoints.

Absolute Setpoints

Parameter name	Values
Comfort setpoint (heating)	0 ... **22** ... 50 (°C)
Standby setpoint (heating)	0 ... **20** ... 50 (°C)
Economy setpoint (heating)	0 ... **18** ... 50 (°C)
Frost protection setpoint	-10 ... **7** ... 15 (°C)
Comfort setpoint (cooling)	0 ... **24** ... 50 (°C)
Standby setpoint (cooling)	0 ... **26** ... 50 (°C)
Economy setpoint (cooling)	0 ... **28** ... 50 (°C)
Overheating protection setpoint	30 ... **35** ... 100 (°C)
Make setpoint changes permanent	**Manually** / Upon mode changes / Never

In the Absolute Setpoints method, setpoints for each operating mode are defined as absolute temperature values. Setpoints for the heating function and for the cooling functions can be defined separately, if the thermostat has been setup to operate in both heating and cooling mode.

Relative Setpoints

Parameter name	Values
Base setpoint after programming	0 ... **23** ... 50 (°C)
User offset (min.)	-10 ... **-5** ... 0 (°C)
User offset (max.)	0 ... **5** ... 10 (°C)
Comfort offset (heating)	-10 ... **-1** ... 0 (°C)
Standby offset (heating)	-10 ... **-3** ... 0 (°C)
Economy offset (heating)	-10 ... **-5** ... 0 (°C)
Frost protection setpoint	-10 ... **7** ... 15 (°C)
Comfort offset (cooling)	0 ... **1** ... 10 (°C)
Standby offset (cooling)	0 ... **3** ... 10 (°C)
Economy offset (cooling)	0 ... **5** ... 10 (°C)
Overheating protection setpoint	30 ... **35** ... 100 (°C)
Reset user offset on operating mode changes	**No** / Yes

In the Relative Setpoint method, setpoints for each operating mode (except for the building protection modes) are defined as offsets relative to a common base setpoint.

3.5.4 Operating Modes

Parameter name	Values
1-bit operating mode objects	**Disabled** / Switch / Trigger
Window status object	Disabled / **Enabled**

The operating mode of the thermostat (comfort, standby, economy, or building protection) can be changed at any time through the *Operating mode* 1-byte communication object (always available), or through four 1-bit communication objects, one per mode (enabled via the *1-bit operating mode objects* parameter). Additionally, the *Window status* communication object can be used to enter a "Forced protection" mode, regardless of the current operating mode selected by the user.

3.5.5 Heating

This section is only shown if the thermostat is setup for heating, or for both heating and cooling.

Parameter name	Values
Control method	**Two-point control** / P-I control

The *Control method* parameter determines the algorithm to use for thermostatic control: Either two-point control with hysteresis (*Two-point control*) or proportional-integral control (*P-I control*). The latter includes both P-I control with continuous output and P-I control with PWM output.

Two-point control

Parameter name	Values
Output type	**1 bit** / 1 byte (0-100%)
Upper hysteresis	1 ... **10** ... 100 (x 0.1ºC)
Lower hysteresis	1 ... **10** ... 100 (x 0.1ºC)
Resending period	**0** ... 255 (x 1 min, 0 = Disabled)

When the two-point control method is selected, on/off commands are sent through the *Output variable (heating)* communication object. This will be a 1-bit object or a 1-byte object, depending on the setting of the *Output type* parameter.

The meaning of the *upper hysteresis* and *lower hysteresis* parameter is described in detail in section 2.4.5 Control algorithms on page 35.

The value of the communication object may be resent to the bus periodically, according to the *Resending period* parameter. Setting this parameter to 0 disables periodic resending.

P-I control

Parameter name	Values
Output type	**Continuous (1 byte)** / PWM (1 bit)
Cycle time	1 ... **15** ... 255 (x 1 min)
Control parameters	Custom parameters
	Warm water heating (5K/150min)
	Floor heating (5K/240min)
	Electric heating (4K/100min)
	Convection heating (4K/90min)
Proportional band *(Only if "custom parameters" is selected)*	1 ... **4** ... 15 (K)
Integral time *(Only if "custom parameters" is selected)*	5 ... **150** ... 255 (x 1 min)
Resending period	**0** ... 255 (x 1 min, 0 = Disabled)

If the proportional-integral (P-I) control method is selected, control commands will be sent through the *Output variable (heating)* communication object. This will be a 1-bit object or a 1-byte object, depending on the setting of the *Output type* parameter.

For proper operation of the P-I control method, the *Cycle time*, *Proportional band*, and *Integral time* parameters need to be defined. The latter two can be defined explicitly, or (more conveniently) one of the available predefined options may be selected from the *Control parameters* list.

See 2.4.5 Control algorithms on page 35 for a detailed description of all control parameters involved.

Additional heating

Parameter name	Values
Additional heating	**No** / Yes
Additional heating band *(Only if additional heating is enabled)*	-100 ... **-25** ... -5 (x 0.1ºC)
Resending period *(Only if additional heating is enabled)*	**0** ... 255 (x 1 min, 0 = Disabled)

These parameters enable control of auxiliary heating systems. For more information, please refer to 2.4.6 Additional heating and cooling on page 40.

3.5.6 Cooling

This section is only shown if the thermostat is setup for cooling, or for both heating and cooling.

Parameter name	Values
Control method	**Two-point control** / P-I control
Output objects *(Only if "Heating and cooling" was selected)*	**Separate heating/cooling objects** / Shared heating/cooling object"

The available configuration parameters and communication objects are equivalent to the parameters and objects described in the **Heating** configuration section, except that they apply to Cooling mode instead.

If the thermostat is setup for both heating and cooling, the additional *Output objects* parameter determines whether separate communication objects will be used for heating and cooling, or whether a shared object will be used for both modes. By default, separate objects are used. If the "shared object" option is selected, all control values are sent through the *Output variable (heating)* communication object.

3.5.7 Simplified configuration mode

In the simplified configuration mode, the operating modes (Comfort, Standby, Economy, and Protection) and all associated communication objects are disabled. The thermostat always operates in a single mode (equivalent to "Comfort"). The *Window status* object is always enabled; when activated, the thermostat enters the special "Forced protection" mode.

Setpoints are always configured as absolute temperature values. Since operating modes are not available, only the *Initial setpoint* (initial setpoint value after an ETS download) and the Protection setpoints need to be configured.

3.6 Inputs

3.6.1 General

Parameter name	Values
Input 1 type	**Disabled**
	Binary - Push button
	Binary - Switch
	Temperature probe
Input 2 type	*Same options as for Input 1*
Input 3 type	*Same options as for Input 1*
Input 4 type	*Same options as for Input 1*

Each of the four available multifunction inputs can be configured as a binary input (in either push button or switch/sensor mode) or as a temperature probe input via the corresponding *Input N type* parameter. Depending on the selection, additional parameters will be enabled. These are described in the following sections.

3.6.2 Binary - Push button

Parameter name	Values
Long press time	1 ... **10** ... 50 (x 0.1 s)
Short press action	**None**
	Send binary values
	Dimmer control
	Shutter control
	Scene control
Long press action	*Same options as for Short press action*
Enable locking object	[Checkbox]

Binary inputs in push button mode can trigger different actions for "short press" and "long press" operations. The *Long press time* parameter defines the minimum press time to differentiate between short and long presses.

The specific actions to be triggered are configured with the *Short press action* and *Long press action* parameters. Additional options for each type of action are described next.

The *Enable locking object* checkbox enables an additional communication object that can be used to lock operation of this input. While the input is locked, short and long presses are ignored, and no actions are triggered.

Send binary values

Parameter name	Values
Send value	**Send 0** / Send 1 / Toggle

If the "Send binary values" action is selected, a binary value will be sent to the bus when the action is triggered. The *Send value* parameter determines the value to send: This can be 0, 1, or "Toggle" (alternates between 0 and 1).

Dimmer control

Parameter name	Values
Operation	**On**
	Off
	Alternate on/off
	Brighter
	Darker
	Alternate brighter/darker
Dimming step	**100%** / 50% / 25% / 12.5% / 6.25% / 3.1% /
(Only for dimming operations)	1.6%

If the "Dimmer control" action is selected, the *Operation* parameter determines the exact operation to perform: A switching operation (on, off, or alternate between on and off) or a dimming operation (brighter, darker, or alternate between brighter and darker).

For dimming operations, the *Dimming step* parameter determines how much should the light intensity be increased or decreased. When the action is triggered, a dimming command will be sent to the bus requesting that the light intensity is adjusted (increased or decreased) by this amount.

Shutter control

Parameter name	Values
Operation	**Move up**
	Move down
	Alternate move up/down
	Step up
	Step down
	Alternate step up/down

If the "Shutter control" action is selected, the *Operation* parameter determines the exact operation to perform: A move operation (move up, move down, or alternate between move up and move down) or a step operation (step up, step down, or alternate between step up and step down).

Scene control

Parameter name	Values
Scene operation	**Recall scene** / Save scene
Scene type	**External** / Internal
Scene number	**1** … 64

If the "Scene control" action is selected, a specific scene will be recalled or saved when the action is triggered, depending on the *Scene operation* parameter. The *Scene type* and *Scene number* parameter select the scene that will be affected by this operation.

3.6.3 Binary - Switch

Parameter name	Values
Action when contact CLOSES	**None** / Send 0 / Send 1 / Toggle 0/1
Action when contact OPENS	*Same options as for Action when contact CLOSES*
Evaluate initial state	**No** / Yes
Enable locking object	[Checkbox]

Binary inputs in switch/sensor mode can trigger actions when the state of the input changes. A different action (send 0, send 1, or toggle between 0 and 1) can be associated to each state (*Action when contact CLOSES* and *Action when contact OPENS*).

Additionally, the *Evaluate initial state* parameter determines whether the state of the input should be evaluated (and the corresponding action triggered) at application startup time, and also after bus voltage recovery.

The *Enable locking object* checkbox enables an additional communication object that can be used to lock operation of this input. While the input is locked, any changes in the input state are ignored, and no actions are triggered.

3.6.4 Temperature probe

Parameter name	Values
Calibration offset	-50 ... **0** ... 50 (x 0.1ºC)
Sending period	**0** ... 255 (x 10 s, 0 = Disabled)
Send when change is greater than	**0** ... 255 (x 0.1ºC, 0 = Disabled)
Alarm function	**Disabled** High temperature alarm Low temperature alarm High/low temperature alarms
High temperature alarm threshold *(If high temperature alarm is enabled)*	-30 ... **30** ... 125 (ºC)
Low temperature alarm threshold *(If low temperature alarm is enabled)*	-30 ... **10** ... 125 (ºC)

The *Calibration offset*, *Sending period*, and *Send when change is greater than* parameters are equivalent to the corresponding parameters for the internal temperature probe (see 3.2.3 Internal temperature sensor on page 48).

If the temperature cannot be read (for example because the temperature probe is broken, or disconnected from the input) an error is signalled through a separate *[INx] Probe error* communication object.

An alarm function can be enable to detect high and/or low temperatures. This is done through the *Alarm function* parameter. Also the thresholds for high and low temperature alarms can be defined. The alarm condition is signalled through the *[INx] Temperature alarm* communication object.

3.7 Scenes

3.7.1 General

Parameter name	Values
Send read requests at startup	No / **Yes**
Scene actuator 1	[Checkbox]
...	
Scene actuator 8	[Checkbox]

Up to 8 scene actuators can be enabled through the corresponding checkboxes. The configuration parameters for each scene actuator are described in the next section.

The parameter *Send read requests at startup* determines whether read requests will be sent at application startup (and after bus voltage recovery) for the actuators' communication objects. See 2.6.4 Storing scenes on page 43.

3.7.2 Scene actuator *N*

Parameter name	Values
Object type	**Switching**
	1-byte unsigned int
	1-byte signed int
	Percentage
	2-byte unsigned int
	2-byte signed int
	2-byte float
	3-byte RGB colour
Scene #1 number	**Disabled** / 1 ... 64
Scene #1 allow saving *(Only if scene not disabled)*	**No** / Yes
Scene #1 initial value *(Only if scene not disabled)*	[See description]
Scene #2	*Same parameters as for scene #1*
...	
Scene #8	*Same parameters as for scene #1*

The *Object type* parameter determines the data type of the communication object for the scene actuator. The following table shows supported data types and the corresponding value ranges.

Data type	Value range
Binary	0 (off) / 1 (on)
1-byte unsigned int	0 … 255
1-byte signed int	-128 … 127
Percentage	0% … 100%
2-byte unsigned int	0 … 65535
2-byte signed int	-32768 … 32767
2-byte float	-671088.64 … 670760.96
3-byte RGB colour	#000000 … #ffffff

Each scene actuator can participate in up to 8 different scenes, identified through the *Scene #X number* parameters.

For each scene in which the actuator is participating, an initial value for the actuator in that scene can be defined (*Scene #X initial value*). This is the value that will be sent to the scene actuator's communication object when the specified scene number is recalled.

Finally, the *Scene #X allow saving* parameter determines whether this value should be updated when a "store request" is received for the specified scene number. See 2.6.4 Storing scenes on page 43 for additional information regarding storing of scenes.

3.8 Door phone (**VERSO** INDOOR)

3.8.1 General

Parameter name	Values
Door station 1	[Checkbox]
...	
Door station 8	[Checkbox]
Additional doorbell	[Checkbox]

Up to 8 door stations can be enabled through the corresponding checkboxes. The configuration parameters for each door station are described in the following section.

The *Additional doorbell* checkbox enables a Doorbell communication object. Writing the value "1" to this object makes the device play a doorbell sound.

3.8.2 Door station *N*

Parameter name	Values
Door phone type	**MURA IP**
	2N IP
	Doorbird
	AUTA ONEX IP SIP
	Fermax MEET
	Generic
Name	[Text field]
Caller ID	[Text field]
Video	**Built-in camera** / Alternative camera
Video stream *(For "Built-in camera")*	**HTTP/MJPEG** RTSP/MJPEG RTSP/MPEG4 RTSP/H.264
Path *(For "Built-in camera")*	[Text field]
Camera *(For "Alternative camera")*	**Camera 1** ... Camera 8

Parameter name	Values
Show video	**Always**
	Only on incoming calls or during calls
	Only during calls
Open doors via	**HTTP** / SIP (DTMF)
DTMF sending method	**SIP INFO** / RFC 2833
(For door opening via DTMF)	
Open door icon	[Icon selection]
Open door HTTP path	[Text field]
(For door opening via HTTP)	
Open door DTMF code	[Text field]
(For door opening via DTMF)	
Enable extra button	[Checkbox]
Extra action icon	[Icon selection]
(If extra button is enabled)	
Extra action HTTP path	[Text field]
(For door opening via HTTP, if extra button is enabled)	
Extra action DTMF code	[Text field]
(For door opening via DTMF if extra button is enabled)	
IP address	**0.0.0.0**
HTTP port	0 … **80** … 65535
RTSP port	0 … **554** … 65535
Username	[Text field]
Password	[Text field]
Enable events	[Checkbox]

General

The *Door phone type* parameter allows selecting the type of door phone. Several presets are available: MURA IP, 2N IP, Doorbird, AUTA, and Fermax MEET. If one of these preset options is selected, some parameters will be configured automatically and will not be shown. For other door phone types, please select *Generic*.

The *Name* parameter is used to identify the door station onscreen in the Door phone page.

If multiple door stations are enabled, the touch panel uses the *Caller ID* parameter to identify the calling station and show the appropriate video stream and door buttons. This parameter

should match the SIP address used by the door station to identify itself in outgoing calls. For more information, please check the door phone documentation.

Video settings

The *Video* parameter allows selecting the video source: either the door station's built-in camera, or an external camera from the ones configured in section 3.9 Cameras (**VERSO** INDOOR) on page 107.

- *Built-in camera*: Depending on the door phone type, you may need to configure the video stream type (*Video stream* parameter) and path (*Path* parameter).
- *Alternative camera*: A camera must be selected through the *Camera* parameter.

Finally, the *Show video* parameter controls when is the video stream displayed:

- *Always:* Video is always shown
- *Only on incoming calls or during calls:* Video is only shown when there is an incoming call (ringing) and during calls
- *Only during calls:* Video is only shown during calls

Door opening

It is possible to control up to two doors per door station. By default, only one door button is shown onscreen. The *Enable extra button* checkbox can be used to enable a secondary button.

The *Open doors via* parameter controls how the door open signals are sent. If *HTTP* is selected, door opening is done by sending **HTTP commands** to the door station. This allows opening the doors even when there is no call in progress. If *SIP (DTMF)* is selected, door opening is done by sending of **DTMF codes** (tones). In the latter case, the *DTMF sending method* parameter determines which method is used to send these codes ("SIP INFO" or "RFC 2833"). For more information, please refer to the door phone documentation.

> **Note**
>
> The following limitations apply when DTMF codes are used:
>
> - DMTF codes can only be sent during calls. If door opening is done using DTMF codes, it will only be possible to open doors during calls.
> - If door opening is done using DTMF codes, the "open door" communication objects will not be available.

Network settings

The door station's networking parameters must also be configured. This includes at least the IP address, and, depending on the door phone type and actual configuration, the HTTP port (for the video stream and door opening commands), RTSP port (for the video stream), and username and password.

Door phone events

The *Enable events* checkbox enables the following 1-bit (DPT 1) communication objects:

- *Incoming call*: When an incoming call is received (ringing), the value "1" (on) will be written to this communication object.

- *Call in progress*: When an incoming call is accepted by the user, the value "1" (on) will be written to this communication object. When the call ends, the value "0" (off) will be written.

- *Open door*: (Only if door opening is done via HTTP). Writing the value "1" to this communication object triggers sending of the HTTP command to open the main door.

- *Trigger extra action*: (Only if door opening is done via HTTP). Writing the value "1" to this communication object triggers sending of the HTTP command to open the secondary door.

- *Door opened*: When the main door is opened from the touch panel, the value "1" (on) will be written to this communication object.

- *Extra action triggered*: When the secondary door is opened from the touch panel, the value "1" (on) will be written to this communication object.

Additional considerations

Please keep in mind the following:

- Some parameters are only available for the "Generic" door phone type; for other door station types, they are configured automatically. These parameters include:
    - Video stream type (except for 2N IP, where this setting is also available)
    - Video stream path
    - HTTP path for door opening commands
    - DTMF sending method
    - HTTP port

- RTSP port

- 2N IP door phones:

    - A * character is automatically appended to any configured DTMF codes
    - Username and password are not used

- Doorbird door phones:

    - A # character is automatically appended to any configured DTMF codes

- Fermax MEET door phones:

    - It is not possible to enable an extra door button
    - The door opening DTMF code is not configurable

- For proper display of the door station's video stream, make sure that the resolution does not exceed 640x480 pixels.

3.9 Cameras (**VERSO** INDOOR)

3.9.1 General

Parameter name	Values
Camera 1	[Checkbox]
...	
Camera 8	[Checkbox]

Up to 8 cameras can be enabled through the corresponding checkboxes. The configuration parameters for each camera are described in the following section.

3.9.2 Camera *N*

Parameter name	Values
Name	[Text field]
Video stream	**HTTP/MJPEG**
	RTSP/MJPEG
	RTSP/MPEG4
	RTSP/H.264
IP address	**0.0.0.0**
HTTP port *(For HTTP/MJPEG video streams)*	0 ... **80** ... 65535
RTSP port *(For RTSP-based video streams)*	0 ... **554** ... 65535
Path	[Text field]
Username	[Text field]
Password	[Text field]

The *Name* parameter is used to identify the camera onscreen in the Cameras page.

The *Video stream* parameter allows selecting the protocol (HTTP or RTSP) and codec (MJPEG, MPEG-4, H.264) of the video stream.

The camera's networking parameters must also be configured. This includes the IP address, HTTP or RTSP port, path, and username and password.

For proper display of the camera's video stream, make sure that the resolution does not exceed 640x480 pixels.

Advanced topics

<div align="right">4</div>

4.1 Product customisation

4.1.1 Background images

The appearance of the user interface of the Iddero Verso touch panel can be customized by using custom background images. Custom background images are uploaded to the device through the USB connection.

Background images should be in PNG or JPEG format. Resolution should be 272x480 pixels in portrait mode, or 480x272 pixels in landscape mode. File names should be as shown in the following table (if using JPEG images, the extension should be `.jpg` or `.jpeg` instead of `.png`):

File name	Affected pages
`home.png`	Home page
`favs.png`	Favorites and Edit favorites pages
`page1.png … page12.png`	Control pages 1 … 12
`default.png`	Default (all pages)

In order to upload the images to the device:

1. Copy the images to the root directory of a FAT32-formatted USB flash drive.

2. Connect the USB flash drive to the device (use the Iddero USB extension cable, part number E-KUSB1).

3. Go to the System settings page, then select USB > Copy images. The new images will be used immediately.

Installed images can be discarded at any time by selecting USB > Discard installed images.

4.1.2 Custom languages

It is possible to install a custom language file in order to translate any system texts to a new language. Please contact Iddero technical support (support@iddero.com) for further information on this topic.

4.2 Firmware updates

If you need to update the firmware of the Iddero Verso touch panel, you can do so through the device's USB connection.

First, you will need to download the firmware release that you want to use. This will typically be a file with `.bin` extension. This file should be copied unmodified to the root directory of a FAT32-formatted USB flash drive. Specifically, do not change the filename, as otherwise it will not be recognized as a valid firmware file.

Connect the USB flash drive to the Iddero Verso touch panel (use the Iddero USB extension cable, part number E-KUSB1) and power on (or reboot) the device.

During the boot process, the USB flash drive will be automatically detected, and the firmware update applied. The whole process takes only a few seconds; the device will beep three times once the update is complete, and will boot with the new firmware automatically.

> **Note**
>
> Iddero Verso will not perform the firmware update if the version of the firmware file found in the USB flash drive matches the version of the currently installed firmware.

You can check the current firmware version at any time in the Info page located in the System Settings menu (see 2.1.6.3 System settings page on page 15).

Remote access

<div style="text-align: right; font-size: 3em;">5</div>

5.1 Introduction

Iddero **VERSO+**IP 2 and **VERSO** INDOOR support remote control of the touch panel from smartphones and tablets through the *Iddero Mobile* app, which is available for Android and iOS. You can download the app directly from Google Play (Android) or from Apple's App Store (iOS).

An idderocloud account is required in order to use the Iddero Mobile app. If you don't have an account yet, you can sign up online through the idderocloud website: www.idderocloud.com.
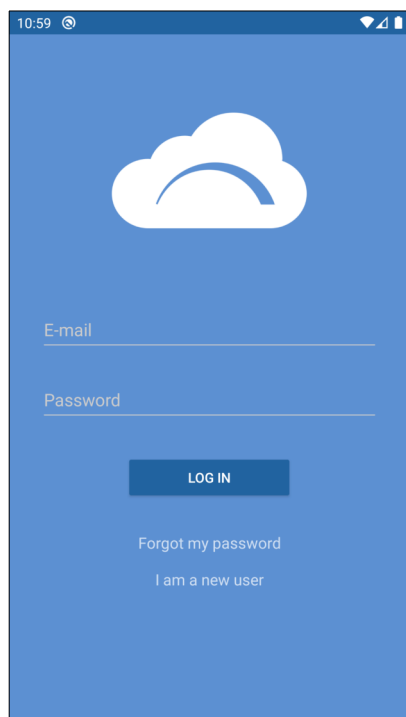
5.2 Iddero Mobile

5.2.1 Overview

When the Iddero Mobile app starts, the Login screen will be initially shown. Enter your idderocloud username and password and tap the "Log in" button.
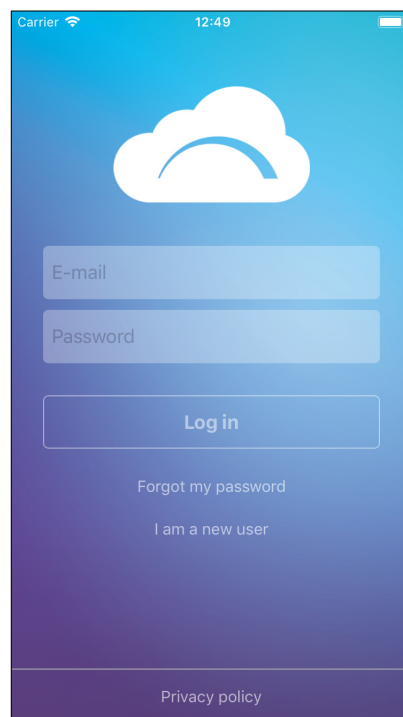
> **Note**
>
> If you don't remember your password, tap "Forgot my password".
>
> If you don't have an idderocloud account yet, tap "I am a new user".
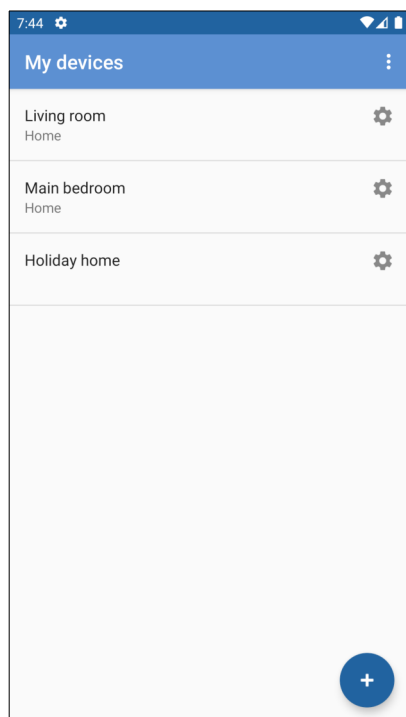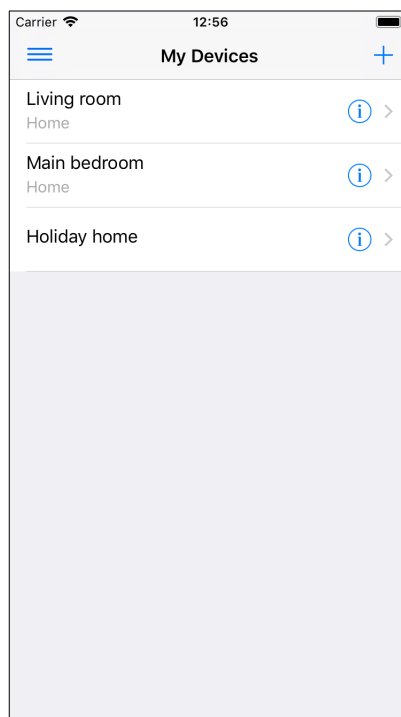
Android   iOS

Once you have successfully logged in, the "My Devices" screen will be shown. This screen lists Iddero devices currently linked to your idderocloud account and compatible with Iddero Mobile.

Android	iOS

- Tap the name of any device to start a remote control session (see 5.2.4 Remote control on page 116).

- Tap the settings icon in the right hand side of each row (Android: ✿, iOS: ⓘ) to adjust device settings (see 5.2.3 Device settings on page 115).

- Tap the Add (+) button to link a new device to your idderocloud account (see 5.2.2 Linking).

You can manually refresh the device list at any time by using a "pull to refresh" gesture: touch anywhere in the device list, then swipe down across the screen. Alternatively, tap the menu button (Android: ⋮ , iOS: ≡), then tap Refresh.

To log out of your account, tap de menu button, then tap "Log out".

5.2.2 Linking

In order to link a **VERSO+**IP 2 or **VERSO** INDOOR device to your idderocloud account:

1. Make sure that both the Iddero device and the smartphone or tablet running the Iddero Mobile app are connected to the Internet.

2. Log into the Iddero Mobile app using your idderocloud username and password.

3. Put the Iddero device in Link mode: go to Settings > System > Remote access, then tap Link.
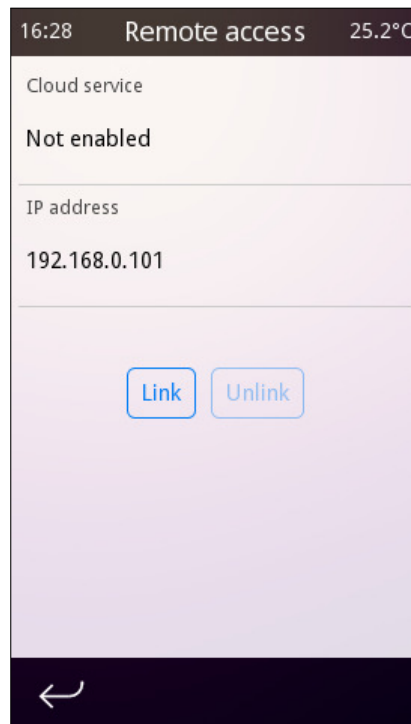
4. In the Iddero Mobile app, tap the Add (+) button and scan the QR code shown in the Iddero touch panel.

5. The linking process will then be performed. Once the operation is complete, the app will return to the "My Devices" screen, and the device list will be updated automatically.

> **Note**
>
> Please note the following:
>
> - Each Iddero device can only be linked to a single idderocloud account at a time. Linking the device to a new account will automatically unlink it from any previous account.
>
> - You can link any number of Iddero devices to the same idderocloud account.
>
> - You can use the same idderocloud account from any number of smartphones and tablets.

5.2.3 Device settings

The Settings screen allows you to adjust some device settings.
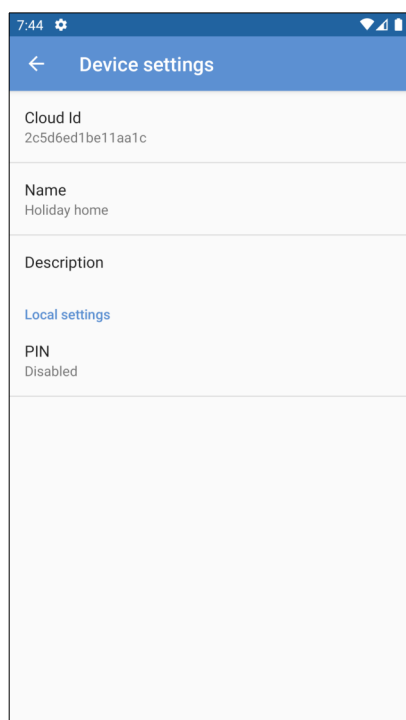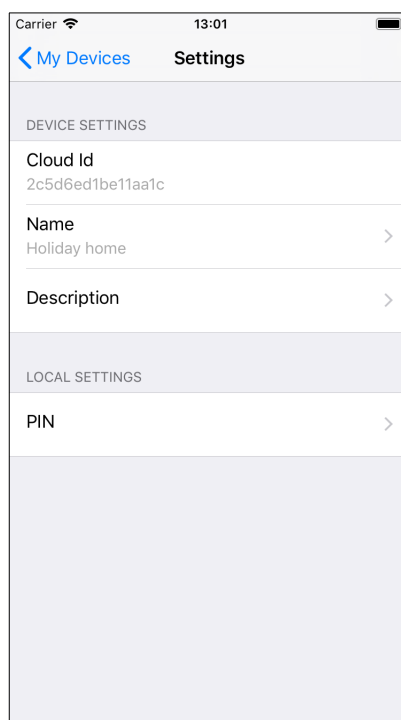
Android   iOS

The following settings are available:

- *Name:* The device name, as shown in the "My Devices" screen.

- *Description:* An optional free-text description of the device.

- *PIN code:* The default PIN code to use for restricted access pages.

The Name and Description parameters are automatically synchronized with your idderocloud account; any change will be visible from all smartphones or tablets where the same idderocloud account is in use. On the other hand, the PIN code field is not synchronized with your idderocloud account: if you enter a default PIN code, it will only take effect on the current smartphone or tablet.

5.2.4 Remote control

When you start a remote control session, the app automatically detects whether it can connect to the Iddero device directly over the local area network (LAN); otherwise, it automatically switches over to cloud-based operation through the idderocloud service.

In both cases, all communications are fully encrypted according to the latest security standards and recommendations.

Once the session has started, the Iddero Mobile app closely mimics the user interface of Iddero touch panel, so no additional learning is required.
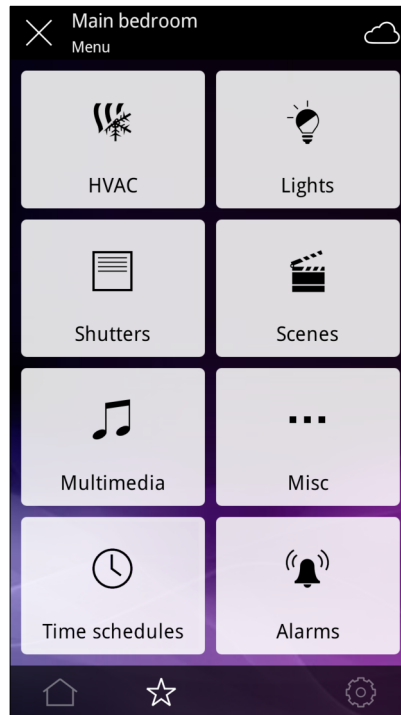
A cloud-shaped icon in the top of the screen (☁) indicates that the connection was established through idderocloud. If this icon is not shown, then the connection is direct (LAN).

Some of the device functions cannot be accessed from the app, or behave in a slightly different way. Specifically:

- The home page is not shown. When a remote control session starts, the app will automatically show the default page, as configured via the *Default page after home page* parameter (see 3.3 User Interface on page 50).

- The title bar does not show the current time and temperature.

- The settings pages are not available.

- The Door phone and Cameras pages are not available.

Tap the ✕ icon on the top of the screen in order to finish the current remote control session and return to the "My Devices" screen.

5.3 Push notifications

VERSO+IP 2 and **VERSO** INDOOR send push notifications automatically when an alarm is triggered. Specifically, push notifications are sent when an alarm switches from the OFF state to the ALARM state (please refer to 2.3 Alarms on page 25).

Push notifications will be received in all smartphones and tablets where the Iddero Mobile app is installed and the user is logged in with the same idderocloud account to which the Iddero device is linked. Notifications are shown automatically onscreen as soon as they are received, even if the Iddero Mobile app is not currently running.

> **Note**
>
> Please bear in mind the following:
>
> - In order to generate push notifications, the Iddero device must be connected to the Internet.
>
> - In order to receive push notifications, the mobile device must be connected to the Internet, and the user must have logged into the Iddero Mobile app using the same idderocloud account to which the Iddero device is linked.
>
> - Both Android and iOS allow users to enable or disable push notifications for any installed app at any time. If you disable push notifications for the Iddero Mobile app, push notifications generated by Iddero devices will not be received.
>
> - Push notifications depend on third-party services such as Apple's Push Notification Service (APNs) and Google's Firebase Cloud Messaging (FCM), which are beyond the control of Iddero.

Annex

6.1 Communication Objects

Miscellaneous

Num.	Name	Type	Flags
1	[Misc] Date	DPT_Date	CT−WU
2	[Misc] Time	DPT_TimeOfDay	CT−WU
3	[Misc] Internal temperature (*)	DPT_Value_Temp	CTR−−
4	[Misc] External temperature	DPT_Value_Temp	CT−WU
5	[Misc] Run/save external scene	DPT_SceneControl	CT−−−
6	[Misc] Wake up	DPT_Trigger	C−−W−
9	[Misc] Timer group 1 lock	DPT_Enable	C−−W−
10	[Misc] Timer group 2 lock	DPT_Enable	C−−W−
11	[Misc] Timer group 3 lock	DPT_Enable	C−−W−
12	[Misc] Timer group 4 lock	DPT_Enable	C−−W−
13	[Misc] Gesture: Up	DPT_Switch	CT−−−
15	[Misc] Gesture: Down	DPT_Switch	CT−−−
17	[Misc] Gesture: Left	DPT_Switch	CT−−−
19	[Misc] Gesture: Right	DPT_Switch	CT−−−
21	[Misc] Gesture: Multitouch	DPT_Switch	CT−−−

(*) Not present in **VERSO** INDOOR.

Components

Num.	Name	Type	Flags
30, 36, 42, 48 … 600	[Cx.y] Binary indicator	DPT_Switch	CT−WU
	[Cx.y] Enumerated indicator	DPT_Value_1_Ucount	CT−WU
	[Cx.y] 1-byte unsigned int indicator	DPT_Value_1_Ucount	CT−WU
	[Cx.y] 1-byte signed int indicator	DPT_Value_1_Count	CT−WU
	[Cx.y] Percentage indicator	DPT_Scaling	CT−WU
	[Cx.y] 2-byte unsigned int indicator	DPT_Value_2_Ucount	CT−WU
	[Cx.y] 2-byte signed int indicator	DPT_Value_2_Count	CT−WU
	[Cx.y] 2-byte float indicator	DPT 9.xxx	CT−WU
	[Cx.y] 4-byte unsigned int indicator	DPT_Value_4_Ucount	CT−WU
	[Cx.y] 4-byte signed int indicator	DPT_Value_4_Count	CT−WU
	[Cx.y] 4-byte float indicator	DPT 14.xxx	CT−WU
	[Cx.y] Dimmer level indicator	DPT_Scaling	CT−WU
	[Cx.y] Shutter position indicator	DPT_Scaling	CT−WU
	[Cx.y] RGB colour	DPT_Colour_RGB	CTRWU
	[Cx.y] RGB red channel	DPT_Scaling	CTRWU
	[Cx.y] Temperature indicator	DPT_Value_Temp	CT−WU
31, 37, 43, 49 … 601	[Cx.y] Binary control	DPT_Switch	CTR−−
	[Cx.y] Enumerated control	DPT_Value_1_Ucount	CTR−−
	[Cx.y] 1-byte unsigned int control	DPT_Value_1_Ucount	CTR−−
	[Cx.y] 1-byte signed int control	DPT_Value_1_Count	CTR−−
	[Cx.y] Percentage control	DPT_Scaling	CTR−−
	[Cx.y] 2-byte unsigned int control	DPT_Value_2_Ucount	CTR−−
	[Cx.y] 2-byte signed int control	DPT_Value_2_Count	CTR−−
	[Cx.y] 2-byte float control	DPT 9.xxx	CTR−−
	[Cx.y] Light on/off	DPT_Switch	CTR−−
	[Cx.y] Shutter step/stop	DPT_Step	CT−−−
	[Cx.y] RGB green channel	DPT_Scaling	CTRWU
	[Cx.y] Temperature control	DPT_Value_Temp	CTR−−
	[Cx.y] Temperature offset control	DPT_Value_Tempd	CTR−−
	[Cx.y] Temperature step control	DPT_Step	CT−−−
	[Cx.y] Alarm trigger	DPT_Alarm	C−−WU
32, 38, 44, 50 … 602	[Cx.y] Relative dimming control	DPT_Control_Dimming	CT−−−
	[Cx.y] Absolute dimming control	DPT_Scaling	CTR−−
	[Cx.y] Shutter up/down	DPT_UpDown	CT−−−
	[Cx.y] Shutter position control	DPT_Scaling	CTR−−
	[Cx.y] RGB blue channel	DPT_Scaling	CTRWU
	[Cx.y] Temperature offset status	DPT_Value_Tempd	CT−WU
	[Cx.y] Alarm acknowledge	DPT_Ack	CT−W−
33, 39, 45, 51 … 603	[Cx.y] White channel	DPT_Scaling	CTRWU

Thermostats

Num.	Name	Type	Flags
606, 641, 815, 850	[Tx] External temperature source	DPT_Value_Temp	Ct-WU
607, 642, 816, 851	[Tx] Heating/Cooling mode	DPT_Heat_Cool	C--W-
608, 643, 817, 852	[Tx] Heating/Cooling mode (status)	DPT_Heat_Cool	CTR--
609, 644, 818, 853	[Tx] On/Off	DPT_Switch	C--W-
610, 645, 819, 854	[Tx] On/Off (status)	DPT_Switch	CTR--
611, 646	[Tx] Operating mode	DPT_HVACMode	C--W-
612, 647	[Tx] Operating mode: Comfort (switch)	DPT_Switch	C--W-
612, 647	[Tx] Operating mode: Comfort (trigger)	DPT_Trigger	C--W-
613, 648	[Tx] Operating mode: Standby (switch)	DPT_Switch	C--W-
613, 648	[Tx] Operating mode: Standby (trigger)	DPT_Trigger	C--W-
614, 649	[Tx] Operating mode: Economy (switch)	DPT_Switch	C--W-
614, 649	[Tx] Operating mode: Economy (trigger)	DPT_Trigger	C--W-
615, 650	[Tx] Operating mode: Protection (switch)	DPT_Switch	C--W-
615, 650	[Tx] Operating mode: Protection (trigger)	DPT_Trigger	C--W-
617, 652, 826, 861	[Tx] Window status	DPT_Window_Door	C--W-
618, 653	[Tx] Operating mode (status)	DPT_HVACMode	CTR--
619, 654, 828, 863	[Tx] Setpoint	DPT_Value_Temp	C--W-
619, 654	[Tx] Base setpoint	DPT_Value_Temp	C--W-
620, 655	[Tx] Base setpoint (status)	DPT_Value_Temp	CTR--
621, 656	[Tx] Setpoint offset	DPT_Value_Tempd	C--W-
622, 657	[Tx] Setpoint step	DPT_Step	CTR--
623, 658	[Tx] Setpoint offset (status)	DPT_Value_Tempd	C--W-
624, 659, 833, 868	[Tx] Setpoint (status)	DPT_Value_Temp	CTR--
625, 660	[Tx] Reset setpoints	DPT_Reset	C--W-
625, 660	[Tx] Reset offset	DPT_Reset	C--W-
626, 661	[Tx] Save setpoint	DPT_Trigger	C--W-
627, 662, 836, 871	[Tx] Output variable (heating)	DPT_Switch	CTR--
627, 662, 836, 871	[Tx] Output variable (heating)	DPT_Scaling	CTR--
628, 663, 837, 872	[Tx] Output variable (cooling)	DPT_Switch	CTR--
628, 663, 837, 872	[Tx] Output variable (cooling)	DPT_Scaling	CTR--
629, 664, 838, 873	[Tx] Aditional heating	DPT_Switch	CTR--
630, 665, 839, 874	[Tx] Aditional cooling	DPT_Switch	CTR--

Multifunction inputs

Num.	Name	Type	Flags
676, 682, 688, 694	[INx] Short press: Binary value	DPT_Switch	CT---
	[INx] Short press: Light on/off	DPT_Switch	CT---
	[INx] Short press: Dimmer control	DPT_Control_Dimming	CT---
	[INx] Short press: Shutter up/down	DPT_UpDown	CT---
	[INx] Short press: Shutter step/stop	DPT_Step	CT---
	[INx] Output value	DPT_Switch	CTR--
	[INx] Current temperature	DPT_Value_Temp	CTR--
677, 683, 689, 695	[INx] Long press: Binary value	DPT_Switch	CT---
	[INx] Long press: Light on/off	DPT_Switch	CT---
	[INx] Long press: Dimmer control	DPT_Control_Dimming	CT---
	[INx] Long press: Shutter up/down	DPT_UpDown	CT---
	[INx] Long press: Shutter step/stop	DPT_Step	CT---
	[INx] Probe error	DPT_Alarm	CTR--
678, 684, 690, 696	[INx] Input lock	DPT_Switch	C--W-
	[INx] Temperature alarm	DPT_Alarm	CTR--

Internal scenes

Num.	Name	Type	Flags
700	[Scenes] Run/save internal scene	DPT_SceneControl	C--W-
701	[Scenes] Actuator 1: Binary value	DPT_Switch	CT-WU
	[Scenes] Actuator 1: 1-byte unsigned int value	DPT_Value_1_Ucount	CT-WU
	[Scenes] Actuator 1: 1-byte signed int value	DPT_Value_1_Count	CT-WU
	[Scenes] Actuator 1: Percentage value	DPT_Scaling	CT-WU
	[Scenes] Actuator 1: 2-byte unsigned int value	DPT_Value_2_Ucount	CT-WU
	[Scenes] Actuator 1: 2-byte signed int value	DPT_Value_2_Count	CT-WU
	[Scenes] Actuator 1: 2-byte float value	DPT 9.xxx	CT-WU
	[Scenes] Actuator 1: RGB value	DPT_Colour_RGB	CT-WU
703	[Scenes] Actuator 2: ...	*(See Actuator 1)*	
705	[Scenes] Actuator 3: ...	*(See Actuator 1)*	
707	[Scenes] Actuator 4: ...	*(See Actuator 1)*	
709	[Scenes] Actuator 5: ...	*(See Actuator 1)*	
711	[Scenes] Actuator 6: ...	*(See Actuator 1)*	
713	[Scenes] Actuator 7: ...	*(See Actuator 1)*	
715	[Scenes] Actuator 8: ...	*(See Actuator 1)*	

*Door phone (**VERSO** INDOOR)*

Num.	Name	Type	Flags
530	[Door phone] Doorbell	DPT_Trigger	C--W-
535	[Door phone] Door station 1: Incoming call	DPT_Trigger	CT---
536	[Door phone] Door station 1: Call in progress	DPT_Trigger	CT---
537	[Door phone] Door station 1: Open door	DPT_Trigger	C--W-
538	[Door phone] Door station 1: Door opened	DPT_Trigger	CT---
539	[Door phone] Door station 1: Trigger extra action	DPT_Trigger	C--W-
540	[Door phone] Door station 1: Extra action triggered	DPT_Trigger	CT---
545 ... 550	[Door phone] Door station 2: ...	*(See Door station 1)*	
555 ... 560	[Door phone] Door station 3: ...	*(See Door station 1)*	
565 ... 570	[Door phone] Door station 4: ...	*(See Door station 1)*	
575 ... 580	[Door phone] Door station 5: ...	*(See Door station 1)*	
585 ... 590	[Door phone] Door station 6: ...	*(See Door station 1)*	
595 ... 600	[Door phone] Door station 7: ...	*(See Door station 1)*	
605 ... 610	[Door phone] Door station 8: ...	*(See Door station 1)*	

IMPORTANT: Only qualified electricians should install, service, or manipulate this equipment. Existing regulations for the prevention of accidents must be observed, as well as any national or local codes and regulations and standard safety precautions.