NAXOS COMBI
Combined KNX/video entryphone panel

NAXOS DOMO
KNX command and display panel

NAXOS COMBI
GW 10 962 WH
GW 12 962 BK

NAXOS DOMO
GW 10 961 WH
GW 12 961 BK

Technical Manual

Contents

1 Introduction

This manual describes the KNX functionalities of the devices:

NAXOS COMBI - Combined KNX/video entryphone panel (GW10962WH - GW12692BK)
NAXOS DOMO - KNX control and display panel (GW10691WH - GW12691BK)
and how they are set and configured with the aid of the ETS configuration software, except the 9.1 paragraph dedicated to the commissioning in Easy mode concerning the Energy management and the Load control.

2 Application

The KNX control panels permit interaction and communication with KNX home and building automation system devices, such as actuators, sensors, dimmers and thermostats for the management of

- Lighting
- Temperature adjustment
- Shutters
- On/off and analogue inputs/outputs.

They also permit the implementation of specific functionalities such as:

- KNX scenes
- Sequence scenes
- Burglar alarm management
- Irrigation
- Energy management
- Load control
- Timers
- Logics
- Malfunction and alarm feedback

The combined KNX/video entryphone panel provides the video entryphone indoor position functionality for the City Vision 2-wire video entryphone system.
In that case, there is interaction between video entryphone events and KNX events, as described below. The KNX Database of the NAXOS DOMO Database does not include this functionality.

ETS makes it possible, in addition to the configuration and parameterisation of the functions and respective communication objects, to set the names of the rooms and zones into which the KNX system is structured, define the names of the control elements and define the access passwords for the rooms and individual elements. Therefore it is not necessary to use an accessory tool for defining the structure of the user interface, which will be automatically generated.

2.1 Association limits

Maximum number of group addresses: 254
Maximum number of associations: 254

This means that up to 254 group addresses can be defined, and up to 254 associations can be made (communication objects and group addresses).

3 *"Main"* Menu

The database of the device for configuration with ETS software allows you to configure the operating parameters, and also gives you the possibility to reconfigure the device with the factory parameters with the Easy controller software operation.

The parameter used to differentiate the two behaviours is **"Programming mode"**. The values that can be set are:

- **ETS mode** **(default value)**
- Easy Controller mode

ETS mode allows the visualisation and configuration of the device operating parameters (S-Mode).

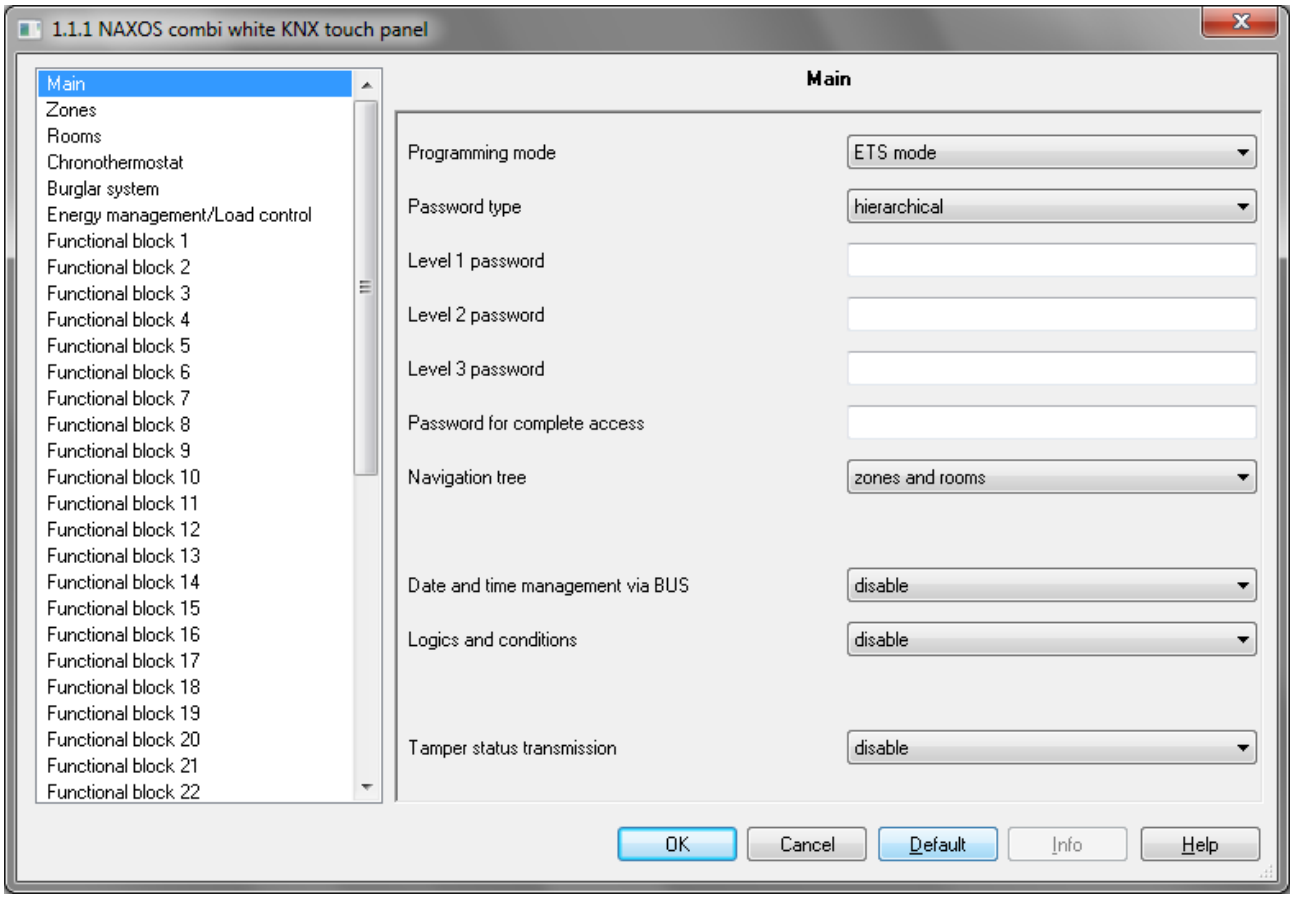


Fig 1: Setting ETS parameters - "Main" section

Navigation in the various panel sections and direct access to the various commands of the individual elements can be protected by a password; there are four access levels available that make it possible to differentiate access to the various panel elements. The method of managing access levels can be:

- **Hierarchical**
  There is a hierarchy between the various access levels; in this mode, entering the password and activating a certain access level also enables access to the other elements with a lower level of protection than the level currently activated
- **Non-hierarchical**
  There is no hierarchy between the various access levels; in this mode, entering the password and activating a certain access level only provides access to the elements related to the activated access level. To access elements with a different access level than what is activated, the relative password must be entered even if the level index is lower than what is already activated.

The **"Password type"** parameter is used to select the access level management method to be adopted. The values that can be set are:

- **hierarchical** **(default value)**
- non-hierarchical

The access levels and relative permissions are shown below:

Type of password	Description
Full access level	Complete access to all elements, even the protected ones. Possibility of local editing (timers, sequence scenes)
Access level 1	Access to elements: ▪ not protected ▪ with access level 1 ▪ in the case of hierarchical management, elements with level 2 and 3 No local editing
Access level 2	Access to elements: ▪ not protected ▪ with access level 2 ▪ in the case of hierarchical management, elements with level 3 No local editing
Access level 3	Access to elements: ▪ not protected ▪ with access level 3 No local editing
Access level without protection	Access to elements: ▪ not protected No local editing

Level 1, in hierarchical management, has higher priority than all the others.
The "full access" level accesses everything even if the selected level management method is "non-hierarchical" and is enabled for the editing of various functions (sequence scenes, timers etc..) which will be described below. The access level activated after entering the relative password remains active until activation of the panel's stand-by mode.

The **"Level 1 password"** parameter is used to enter up to 8 alphanumeric characters that identify the password for accessing elements with access level 1; the values that can be set are:

- maximum 8 alphanumeric characters

The **"Level 2 password"** parameter is used to enter up to 8 alphanumeric characters that identify the password for accessing elements with access level 2; the values that can be set are:

- maximum 8 alphanumeric characters

The **"Level 3 password"** parameter is used to enter up to 8 alphanumeric characters that identify the password for accessing elements with access level 3; the values that can be set are:

- maximum 8 alphanumeric characters

The **"Password for complete access"** parameter is used to enter up to 8 alphanumeric characters that identify the "full access" password; the values that can be set are:

- maximum 8 alphanumeric characters

The touch panel automatically manages the internal calendar, updating both the date and the time shown on the display. However, it is possible to enable date and time management via BUS, reserving communication objects both for receiving the date and time from the KNX BUS as well as for sending the date and time update on the KNX BUS.
The **"Date and time management via BUS"** parameter is used to enable date and time management via KNX BUS; the values that can be set are:

- **disable** **(default value)**
- enable

By selecting **enable**, the **Functional block 15** menu is not displayed and the associated communication objects are reserved for date and time management;
the **"Request time and date update when power supply voltage is restored"** and **"Send date and time update on KNX BUS"** parameters are displayed together with the communication objects ***Date and Time - Date input*** (Date Point Type: 11.001 DPT_Date), ***Date and Time - Date output*** (Date Point Type: 11.001 DPT_Date), ***Date and Time - Time of day input*** (Date Point Type: 10.001 DPT_TimeOfDay), ***Date and Time - Time of day output*** (Date Point Type: 10.001 DPT_TimeOfDay), ***Date and Time - Daylight saving time autom. update*** (Date Point Type: 1.001 DPT_Switch) and ***Date and Time - Daylight saving time output*** (Date Point Type: 1.001 DPT_Switch).

The **"Request time and date update when power supply voltage is restored"** parameter is used to enable request date and time update when power supply voltage is restored for the device; the values that can be set are:

- **disable** **(default value)**
- enable

The **"Send date and time update on KNX BUS"** parameter is used to set the conditions for sending the current date and day/time to the other KNX system devices to keep them synchronised with the panel; the values that can be set are:

- **after blackout** **(default value)**
- after a modification
- after a blackout or a modification
- after a blackout, modification and periodically

The "after blackout" sending condition involves sending the BUS telegrams with the current time/day and date usually after the BUS or auxiliary power supply voltage is restored.
The "after a modification" sending condition involves sending BUS telegrams after changing the date or day/time on the panel, regardless of the fact that the change took place using the local navigation menu or via BUS telegram on the communication objects ***Date and Time - Time of day input*** and ***Date and Time - Date input***.
The "periodic" sending condition involves the systematic sending of the date and time after a certain period of time has passed since the last transmission; this setting displays the **"Update period"** parameter, used to define the period for sending the date and day/time update telegrams. The values that can be set are:

- **every 6 hours** **(default value)**
- every 12 hours
- every 24 hours
- every 7 days

Navigation in the "Rooms" section of the panel takes place on two levels: Zones and rooms. Depending on the structure of the building where the panel is installed, it may be sufficient to have a single zone that includes all the rooms; for easy ETS configuration using the **"Navigation tree"** parameter, it is possible to define if the structure should maintain two levels (zones and rooms) or only one level. The values that can be set are:

- rooms
- **zones and rooms** **(default value)**

The panel can perform actions depending on the result of the logic operations or comparisons that are programmed by ETS; these functions can be enabled using the **"Logics and conditions"** parameter, which may assume the following values:

- **disabled** **(default value)**
- enable

selecting **enable** displays the **Logics and conditions** menu, which will be described below.

The parameter **"Tamper status transmission"** is used to enable a communication object dedicated to reporting the status of the tamper located on the back of the device; this information could be used, for example, to report a manumission/removing of the panel from its support. The possible values are:

- **disable** **(default value)**
- enable

selecting the value **enable**, become visible the parameters **"Feedback associated with tamper status"** and **"Tamper status repetition time"** and the communication object ***Tamper status feedback*** (Data Point Type: 1.001 DPT_Switch).
The parameter **"Feedback associated with tamper status"** is used to assign the value that is reported on the KNX bus with the tamper status (open/closed); the possible values are:

- **only contact closed -> 1** **(default value)**
- only contact closed -> 0
- only contact open -> 1
- only contact open -> 0
- contact closed -> 1 / contact open -> 0
- contact open -> 1 / contact closed -> 0

The feedback status message is sent via the object ***Tamper status feedback*** after a voltage restore, on request and on tamper status change; it is possible to enable and select the period of the status telegram cyclic repetition via the parameter **"Tamper status repetition time"**, which can assume the following values:

- from **no repetition (default value)** to 60 minutes with step of 1

4 "*Zones*" menu

This menu contains the parameters used to customise the names and access levels for the 8 zones available, for the creation of a navigation tree.

The structure of the menu is as follows:

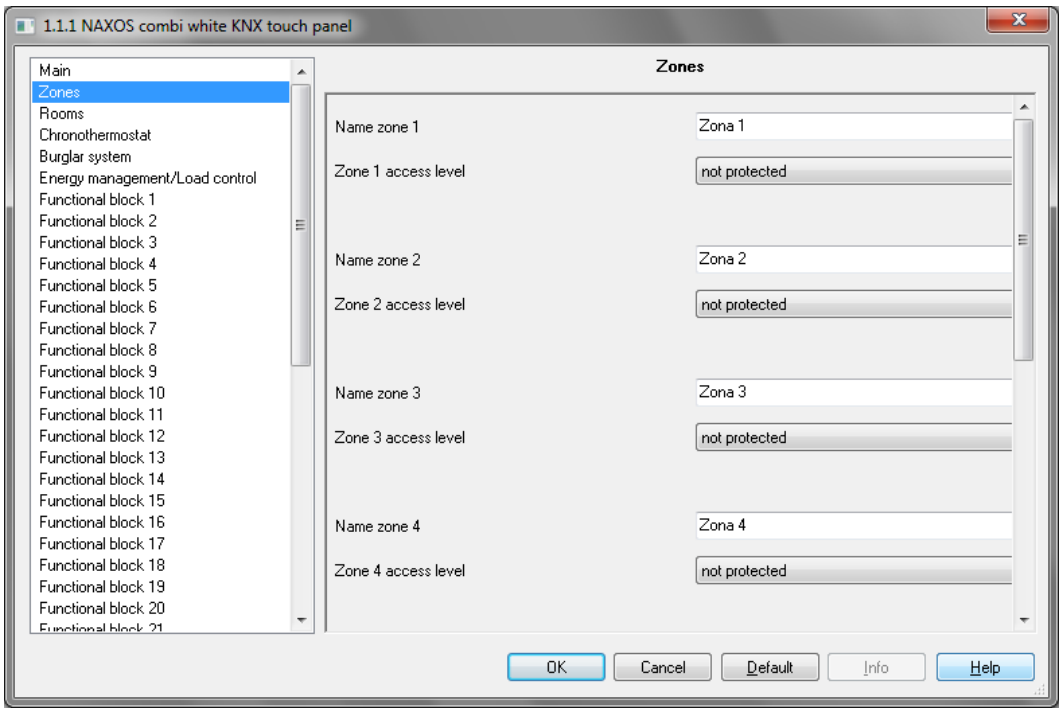


Fig 2: Setting ETS parameters - "Zones" section

With the parameters **"Zone i name"** (1 ≤ i ≤ 8), which are displayed if in the item **"Navigation tree"** in the **Main** menu the **zones and rooms** value is set, it is possible to enter the name you want to associate with a certain zone "i" displayed in the navigation structure; the possible values are:

- maximum 20 alphanumeric characters

The **"Room name"** parameter, which is displayed if in the item **"Navigation tree"** in the **Main** menu the **rooms** value is set, it is possible to enter the name you want to associate with the system, as in this case the navigation structure is only on one level; the possible values are:

- maximum 20 alphanumeric characters

With the parameters **"Zone i access level"** (1 ≤ i ≤ 8), which are displayed if in the item **"Navigation tree"** in the **Main** menu the **zones and rooms** value is set, it is possible to define the rights required to access the "i-th" zone; the possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

The parameter **"Access level"**, which is displayed if in the item **"Navigation tree"** in the **Main** menu, the **only rooms** value is set, it is possible to define the rights required to access the room sections of the panel; the possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

The zones that do not include rooms will not be displayed in the navigation structure.

5 "*Rooms*" menu

This menu contains the parameters used to customise the names, the associated zones and the access levels for the 32 rooms available, for the creation of a navigation tree.

The structure of the menu is as follows:

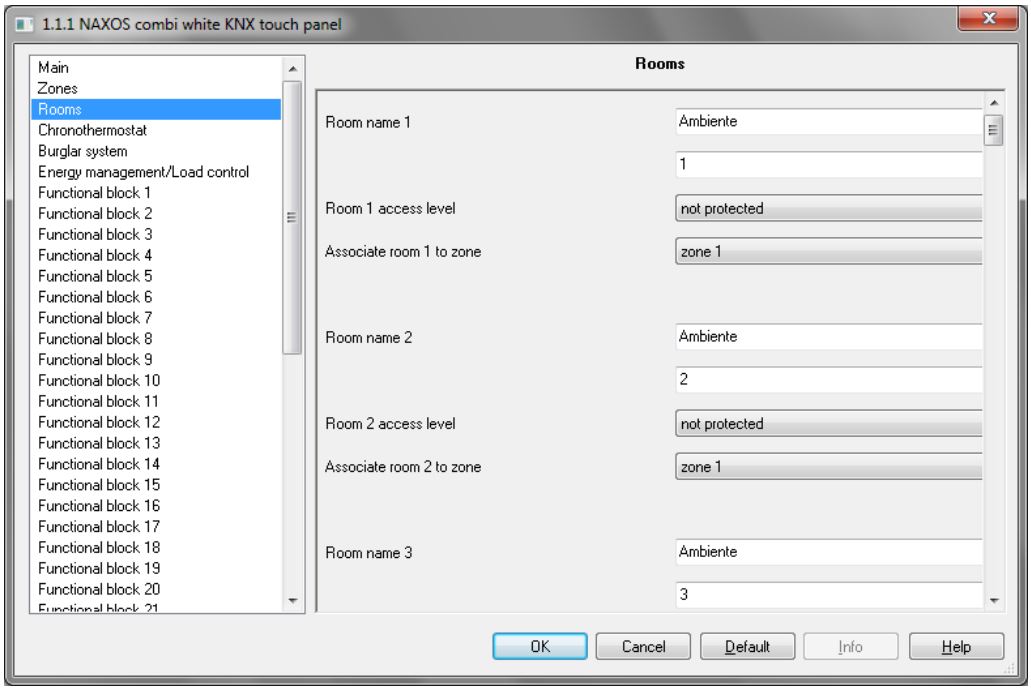


Fig 3: Setting ETS parameters - "Rooms" section

With the parameters **"Room i name"** (1 ≤ i ≤ 32) it is possible to enter the name you want to associate with a certain room "i" displayed in the navigation structure; the name consists of two rows, as the room is

represented on a graphical level inside tiles with defined dimensions. When the room name is instead displayed in the title of the navigation page, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

With the parameters **"Room i access level"** (1 ≤ i ≤ 32) it is possible to define the rights required to access the "i-th" zone; the possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the 2-level navigation structure, each Room must be associated with a Zone; this association is defined with the parameters **"Associate room i with zone"** (1 ≤ i ≤ 32), which can have the following values:

- **Zone 1** **(default value)**
- Zone 2
- Zone 3
- Zone 4
- Zone 5
- Zone 6
- Zone 7
- Zone 8

The rooms that do not include elements will not be displayed in the navigation structure.

6 "*Functional block x*" menu

The panel's communication objects are divided into a total of 42 functional blocks
A dedicated menu called **Functional block x** (1 ≤ i ≤ 42) is displayed for each available functional block.
The menu structure changes based on the value set for the **"Associated function"** parameter; each functional block groups 6 communication objects, whose function also depends on the Associated function selected.
For the sake of simplicity, the parameters enabled according to the value set for the above parameter are listed in the following paragraphs.

The basic structure of the menu is as follows:

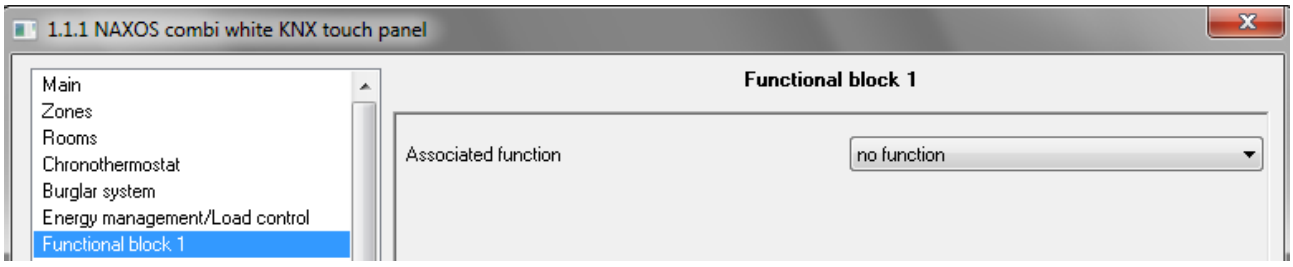


Fig 4: Setting ETS parameters - "Functional block x" section

The parameter **"Associated function"** determines the function matches with the generic functional block x; based on the value set for this item, the Functional block x menu will appear differently. The values that can be set are:

- **no function** **(default value)**
- 1 dimmer
  (See paragraph “1 dimmer” function)
- 2 dimmers
  (See paragraph “2 dimmers” function)
  3 dimmers
  (See paragraph “3 dimmers” function)
- DALI dimmer
  (See paragraph “DALI dimmer” function)
- RGB dimmer
  (See paragraph “RGB dimmer” function)
- 3 RGB dimmers
  (See paragraph “3 RGB dimmers” function)
- 1 light
  (See paragraph “1 light” function)
- 2 lights
  (See paragraph “2 lights” function)
- 3 lights
  (See paragraph “3 lights” function)
- 1 on/off actuator
  (See paragraph “1 on/off actuator” function)
- 2 on/off actuators
  (See paragraph “2 on/off actuators” function)
- 3 on/off actuators
  (See paragraph “3 on/off actuators” function)
- 1 shutter
  (See paragraph “1 shutter” function)
- 2 shutters
  (See paragraph “2 shutters” function)
- 3 shutters
  (See paragraph “3 shutters” function)
- HVAC master
  (See paragraph “HVAC master” function)
- Irrigation
  (See paragraph “Irrigation” function)
- 6 independent inputs
  (See paragraph “6 independent inputs” function)
- 6 independent outputs
  (See paragraph “6 independent outputs” function)
- KNX scenes
  (See paragraph “KNX scenes” function)
- 6 videoentryphone events
  (See paragraph “6 videoentryphone events” function)

All blocks can be configured based on the standard functions listed above. Each functional block uses 6 KNX communication objects.
In addition to the functions common to all blocks, it is possible to configure blocks 15 to 42 to carry out dedicated functions, which are outlined below:

Block (communication objects)	Dedicated functions alternatives to the standard functions
15 (84,85,86,87,88,89)	Date and time
16 (90,91,92,93,94,95)	Chronothermostat zone 1
17 (96,97,98,99,100,101)	
18 (102,103,104,105,106,107)	Heating fancoil zone 1
19 (108,109,110,111,112,113)	Air conditioning fancoil zone 1
20 (114,115,116,117,118,119)	Chronothermostat zone 2
21 (120,121,122,123,124,125)	
22 (126,127,128,129,130,131)	Heating fancoil zone 2
23 (132,133,134,135,136,137)	Air conditioning fancoil zone 2
24 (138,139,140,141,142,143)	Chronothermostat zone 3
25 (144,145,146,147,148,149)	
26 (150,151,152,153,154,155)	Heating fancoil zone 3
27 (156,157,158,159,160,161)	Air conditioning fancoil zone 3
28 (162,163,164,165,166,167)	Chronothermostat zone 4
29 (168,169,170,171,172,173)	
30 (174,175,176,177,178,179)	Heating fancoil zone 4
31 (180,181,182,183,184,185)	Air conditioning fancoil zone 4
32 (186,187,188,189,190,191)	Burglar alarm control area 4
33 (192,193,194,195,196,197)	
34 (198,199,200,201,202,203)	Burglar alarm control area 3
35 (204,205,206,207,208,209)	
36 (210,211,212,213,214,215)	Burglar alarm control area 2
37 (216,217,218,219,220,221)	
38 (222,223,224,225,226,227)	Burglar alarm control area 1
39 (228,229,230,231,232,233)	
40 (234,235,236,237,238,239)	Total burglar alarm command and alarm feedback
41 (240,241,242,243,244,245)	
42 (246,247,248,249,250,251)	Load control/Energy management

By enabling one of the dedicated functions, the relative operating blocks used will no longer be available in the configuration menu to carry out standard functions.

6.1 "1 dimmer" function

This function is used to create a graphic element for controlling a dimmer with all the main control components.
The following commands are always available:

- ON/OFF managed by the object ***Block x - Dimmer switching*** (Date Point Type: 1.001 DPT_Switch)
- relative brightness regulation (100% brightness increase/decrease and regulation stop command) managed by means of the object ***Block x - Dimmer brightness regulation*** (Date Point Type: 3.007 DPT_Control_Dimming)
- absolute brightness regulation (brightness % value setting) managed by means of the object ***Block x - Dimmer brightness value command*** (Date Point Type: 5.001 DPT_Scaling)

In the same way, the graphic element is always able to display:

- the ON or OFF status of the dimmer, received from the BUS via the object ***Block x - Dimmer status feedback*** (Date Point Type: 1.001 DPT_Switch)
- the current brightness level of the dimmer, received from the BUS via the object ***Block x - Dimmer brightness value feedback*** (Date Point Type: 5.001 DPT_Scaling)

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block x - Dimmer status feedback*** and ***Block x - Dimmer brightness value feedback*** to update the status information.
The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.

The basic structure of the menu is as follows:

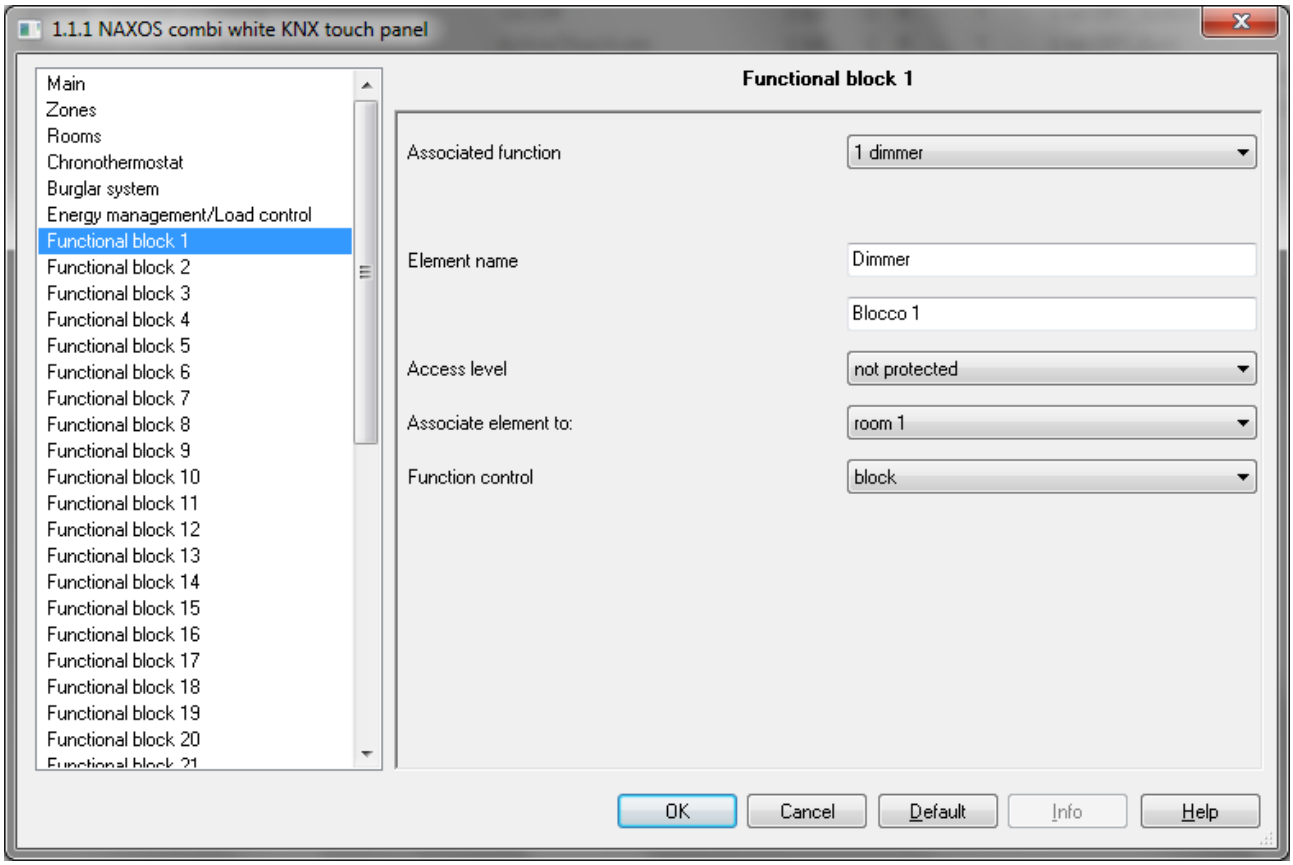


Fig 5: Setting ETS parameters - "1 dimmer function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 1 dimmer element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Function control"** is used to expand the range of commands for controlling the dimmer; in addition to the commands that were already mentioned and always enabled, the 1 dimmer element can be configured to send block on/off commands or priority commands (forcing).
The values that can be set are:

- **block** **0 (default value)**
- forcing 1

selecting the value **block** displays the communication object ***Block x - Dimmer block*** (Date Point Type: 1.002 DPT_Bool); vice versa, selecting the value **forcing** displays the communication object ***Block x - Dimmer priority command*** (Date Point Type: 2.001 DPT_Switch_Control).

6.2 "2 dimmers" function

This function is used to create two graphic elements for controlling two dimmers with some of the main control components.
The following command becomes available:

- ON/OFF managed by the object ***Block xA - Dimmer switching*** (Date Point Type: 1.001 DPT_Switch) for the first element and ***Block xB - Dimmer switching*** (Date Point Type: 1.001 DPT_Switch) for the second

The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.

The basic structure of the menu is as follows:

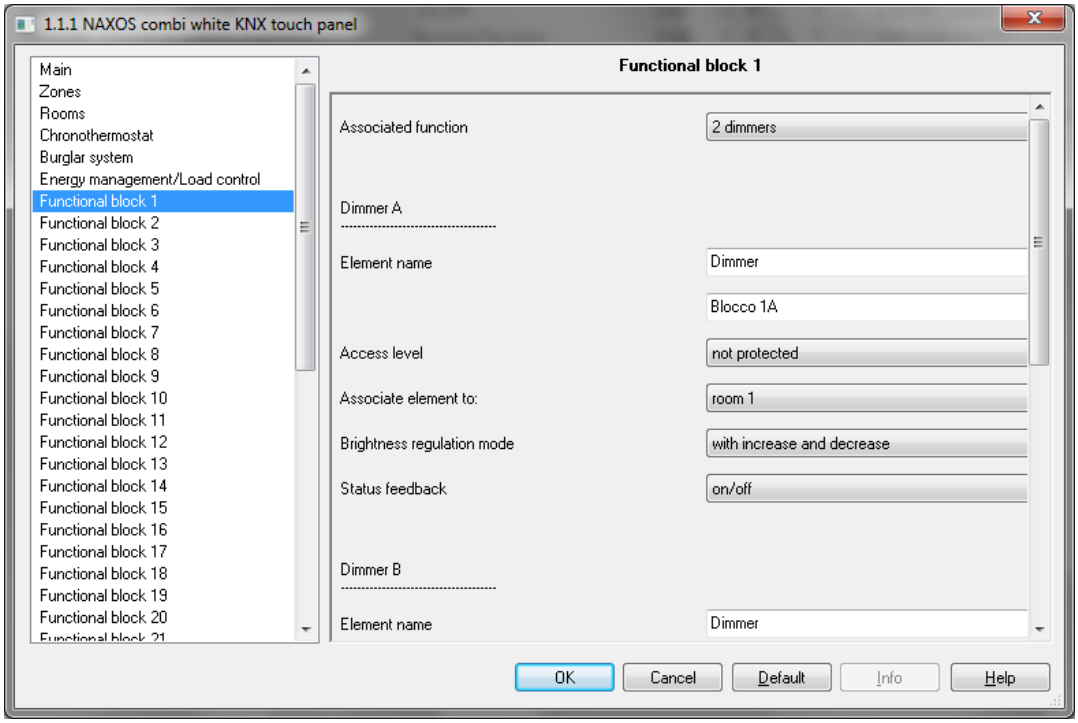


Fig 6: Setting ETS parameters - "2 dimmers function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 2 dimmer element (A or B, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A or B, depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A or B, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Brightness regulation mode"** is used to define the method for controlling the brightness of the element (A or B, depending on the header to which the parameter refers) for dimmer control; in addition to the commands that were already mentioned and always enabled, it is possible to configure the 2 dimmer element to send relative brightness regulation commands or absolute brightness regulation commands.
The values that can be set are:

- **with increase and decrease** **(default value)**
- with sending of percentage value

selecting the value **with increase and decrease** displays the communication object ***Block xA - Dimmer brightness regulation*** (Date Point Type: 3.007 DPT_Control_Dimming) if the object refers to the first element or ***Block xB - Dimmer brightness regulation*** (Date Point Type: 3.007 DPT_Control_Dimming) if the object refers to the second; vice versa, selecting the value **with sending of percentage value**, displays the communication object ***Block xA - Dimmer brightness value command*** **(**Date Point Type: 5.001 DPT_Scaling) if the object refers to the first element or ***Block xB - Dimmer brightness value command*** (Date Point Type: 5.001 DPT_Scaling) if the object refers to the second.

The parameter **"Status feedback"** is used to define the type of status information of the element (A or B, depending on the header to which the parameter refers) you want to display.
The values that can be set are:

- **on/off** **0 (default value)**
- % value 1

selecting the value **on/off** displays the communication object ***Block xA - Dimmer status feedback*** (Date Point Type: 1.001 DPT_Switch) if the object refers to the first element or ***Block xB - Dimmer status feedback*** (Date Point Type: 1.001 DPT_Switch) if the object refers to the second; vice versa, selecting **% value**, displays the communication object ***Block xA - Dimmer brightness value feedback*** (Date Point Type: 5.001 DPT_Scaling) if the object refers to the first element or **Block xB - Dimmer brightness value feedback** (Date Point Type: 5.001 DPT_Scaling) if the object refers to the second.
When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects enabled for updating the status information.

6.3 *"3 dimmers"* function

This function is used to create three graphic elements for controlling three dimmers with some of the main control components.

The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.

The basic structure of the menu is as follows:

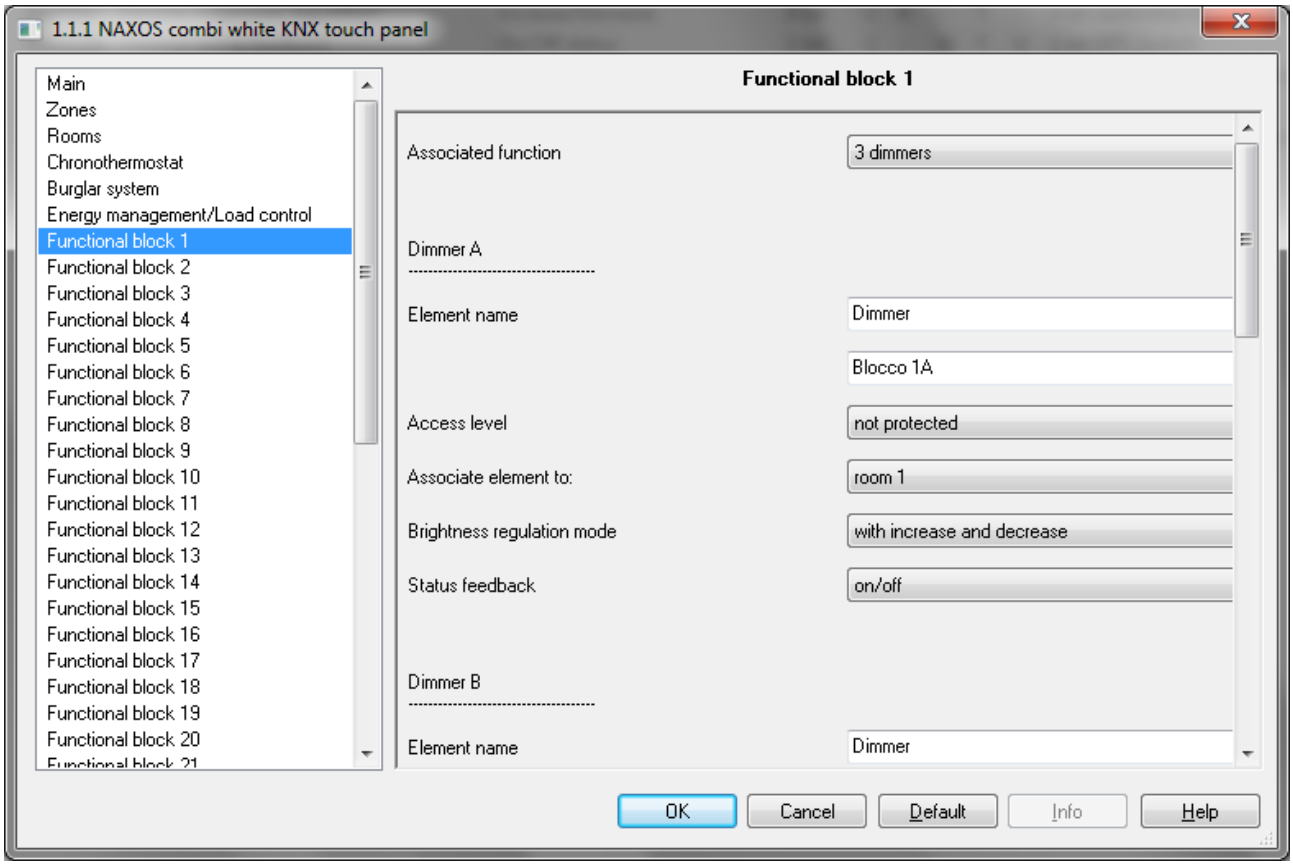


Fig 7: Setting ETS parameters - "3 dimmers function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 3 dimmer element (A, B or C, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A, B or C depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A, B or C, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2

- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Brightness regulation mode"** is used to define the method for controlling the brightness of the element (A, B or C, depending on the header to which the parameter refers) for dimmer control; in addition to the commands that were already mentioned and always enabled, it is possible to configure the 2 dimmer element to send relative brightness regulation commands or absolute brightness regulation commands.
The values that can be set are:

- **with increase and decrease (default value)**
- with sending of percentage value

selecting the value **with increase and decrease** displays the communication object ***Block xA - Dimmer brightness regulation*** (Date Point Type: 3.007 DPT_Control_Dimming) if the object refers to the first element, ***Block xB - Dimmer brightness regulation*** (Date Point Type: 3.007 DPT_Control_Dimming) if the object refers to the second element or ***Block xC - Dimmer brightness regulation*** (Date Point Type: 3.007 DPT_Control_Dimming) if the object refers to the third; vice versa, selecting the value **with sending of percentage value**, displays the communication object ***Block xA - Dimmer brightness value command*** (Date Point Type: 5.001 DPT_Scaling) the object refers to the first element, ***Block xB - Dimmer brightness value command*** (Date Point Type: 5.001 DPT_Scaling) if the object refers to the second element or ***Block xC - Dimmer brightness value command*** (Date Point Type: 5.001 DPT_Scaling) if the object refers to the third.

The parameter **"Status feedback"** is used to define the type of status information of the element (A, B or C, depending on the header to which the parameter refers) you want to display.
The values that can be set are:

- **on/off (default value)**
- % value

selecting the value **on/off** displays the communication object ***Block xA - Dimmer status feedback*** (Date Point Type: 1.001 DPT_Switch) if the object refers to the first element, **Block xB - Dimmer status feedback** (Date Point Type: 1.001 DPT_Switch) if the object refers to the second element or ***Block xC - Dimmer status feedback*** (Date Point Type: 1.001 DPT_Switch) if the object refers to the third; vice versa, selecting **% value**, displays the communication object ***Block xA - Dimmer brightness value feedback*** (Date Point Type: 5.001 DPT_Scaling) the object refers to the first element, ***Block xB - Dimmer brightness value feedback*** (Date Point Type: 5.001 DPT_Scaling) if the object refers to the second element or ***Block xC - Dimmer brightness value feedback*** (Date Point Type: 5.001 DPT_Scaling) if the object refers to the third.
When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the enabled objects to update the status information.

6.4 "DALI dimmer" function

This function is used to create a graphic element for controlling a DALI dimmer with all the main control components.
The following commands are always available:

- ON/OFF managed by the object ***Block x - DALI dimmer/group switching*** (Date Point Type: 1.001 DPT_Switch)
- relative brightness regulation (100% brightness increase/decrease and regulation stop command) managed by means of the object ***Block x - DALI dimmer/group brightness regulation*** (Date Point Type: 3.007 DPT_Control_Dimming)
- absolute brightness regulation (brightness % value setting) managed by means of the object ***Block x - DALI dimmer/group brightness command*** (Date Point Type: 5.001 DPT_Scaling)

In the same way, the graphic element is always able to display:
- the ON or OFF status of the dimmer, received from the BUS via the object ***Block x - DALI dimmer/group status feedback*** **(**Date Point Type: 1.001 DPT_Switch)
- the current brightness level of the dimmer, received from the BUS via the object ***Block x - DALI dimmer/group brightness feedback*** (Date Point Type: 5.001 DPT_Scaling)
- alarm/fault condition of the ballast or the lamp connected to the dimmer received from the BUS via the object ***Block x - DALI dimmer/group error feedback*** (Date Point Type: 1.005 DPT_Alarm)

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block x - DALI dimmer/group status feedback, Block x - DALI dimmer/group brightness feedback*** and ***Block x - DALI dimmer/group error feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.
The basic structure of the menu is as follows:

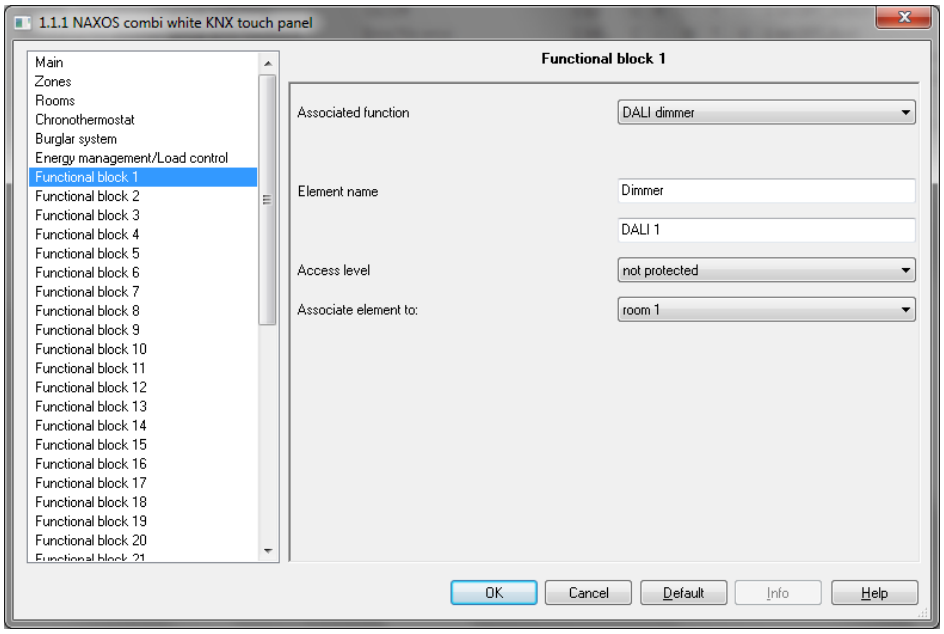


Fig 8: Setting ETS parameters - "DALI dimmer function" section

The parameter **"Element name"** is used to enter the name you want to associate with the DALI dimmer element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:
- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:
- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:
- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

6.5 *"RGB dimmer"* function

This function is used to create a graphic element for controlling an RGB dimmer with the commands for controlling all the colour components.
The following commands are always available:

- absolute brightness regulation (brightness % value setting) of the RED colour component managed by means of the object ***Block x - RGB red brightness value command*** (Date Point Type: 5.010 DPT_Value_1_Ucount)
- absolute brightness regulation (brightness % value setting) of the GREEN colour component managed by means of the object ***Block x - RGB green brightness value command*** (Date Point Type: 5.010 DPT_Value_1_Ucount)
- absolute brightness regulation (brightness % value setting) of the BLUE colour component managed by means of the object ***Block x - RGB blue brightness value command*** (Date Point Type: 5.010 DPT_Value_1_Ucount)

In the same way, the graphic element is always able to display:

- the current brightness level of the RED colour component, received from the BUS via the object ***Block x - RGB red brightness value feedback*** (Date Point Type: 5.010 DPT_Value_1_Ucount)
- the current brightness level of the GREEN colour component, received from the BUS via the object ***Block x - RGB green brightness value feedback*** (Date Point Type: 5.010 DPT_Value_1_Ucount)
- the current brightness level of the BLUE colour component, received from the BUS via the object ***Block x - RGB blue brightness value feedback*** (Date Point Type: 5.010 DPT_Value_1_Ucount)

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block x - RGB red brightness value feedback***, ***Block x - RGB green brightness value feedback*** e ***Block x - RGB blue brightness value feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.

The basic structure of the menu is as follows:

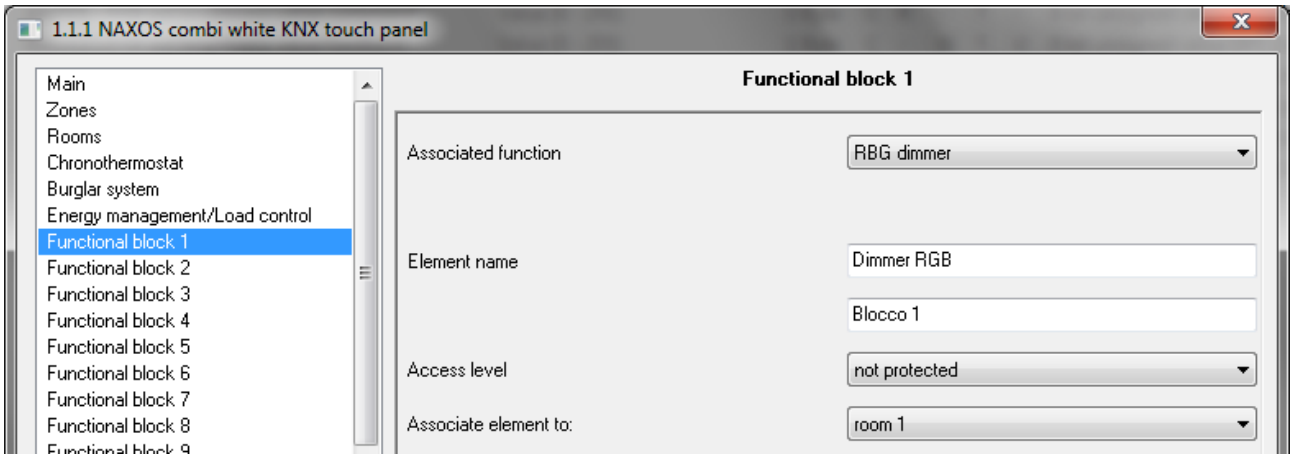


Fig 9: Setting ETS parameters - "RGB dimmer function" section

The parameter **"Element name"** is used to enter the name you want to associate with the RGB dimmer element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

6.6 "3 RGB dimmers" function

This function is used to create three graphic elements for controlling three RGB dimmers with the commands for controlling all the colour components.
The following command becomes available:

- absolute brightness regulation (brightness % value setting) of the RED, GREEN and BLUE colour components managed by means of the object ***Block xA - RGB brightness value command*** (Date Point Type: 232.600 DPT_Colour_RGB) for the first element, ***Block xB - RGB brightness value command*** (Date Point Type: 232.600 DPT_Colour_RGB) for the second and ***Block xC - RGB brightness value command*** (Date Point Type: 232.600 DPT_Colour_RGB) for the third

In the same way, the graphic element is always able to display:

- the current brightness level of the RED, GREEN and BLUE colour components, received from the BUS via the object ***Block xA - RGB brightness value feedback*** (Date Point Type: 232.600 DPT_Colour_RGB) for the first element, ***Block xB - RGB brightness value feedback*** (Date Point Type: 232.600 DPT_Colour_RGB) for the second and ***Block xC - RGB brightness value feedback*** (Date Point Type: 232.600 DPT_Colour_RGB) for the third

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block xA - RGB brightness value feedback***, ***Block xB - RGB brightness value feedback*** and ***Block xC - RGB brightness value feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.

The basic structure of the menu is as follows:

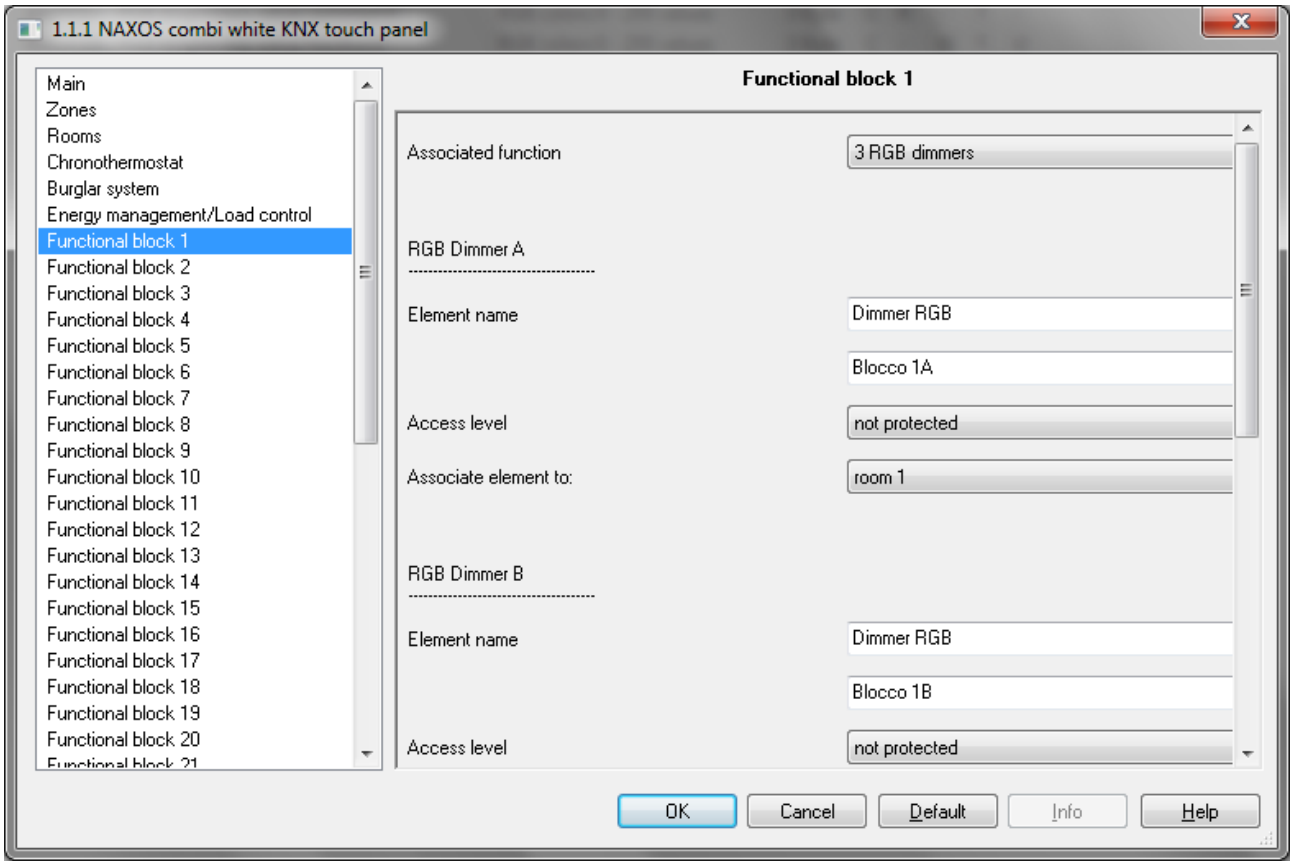


Fig 10: Setting ETS parameters - "3 RGB dimmers function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 3 RGB dimmer element (A, B or C, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A, B or C depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A, B or C, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

6.7 *"1 light"* function

This function is used to create a graphic element for controlling one light with all the main control components.
The following commands are always available:

- ON/OFF managed by the object ***Block x - Light switching*** (Date Point Type: 1.001 DPT_Switch)
- on/off forcing and forcing deactivation (priority commands) managed via the object ***Block x - Light priority command*** (Date Point Type: 2.001 DPT_Switch_Control)
- block function on/off managed via the object ***Block x - Light block*** (Date Point Type: 1.002 DPT_Bool)

In the same way, the graphic element is always able to display:

- the ON or OFF status of the light, received from the BUS via the object ***Block x - Light status feedback*** (Date Point Type: 1.001 DPT_Switch)

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the object ***Block x - Light status feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.

The basic structure of the menu is as follows:

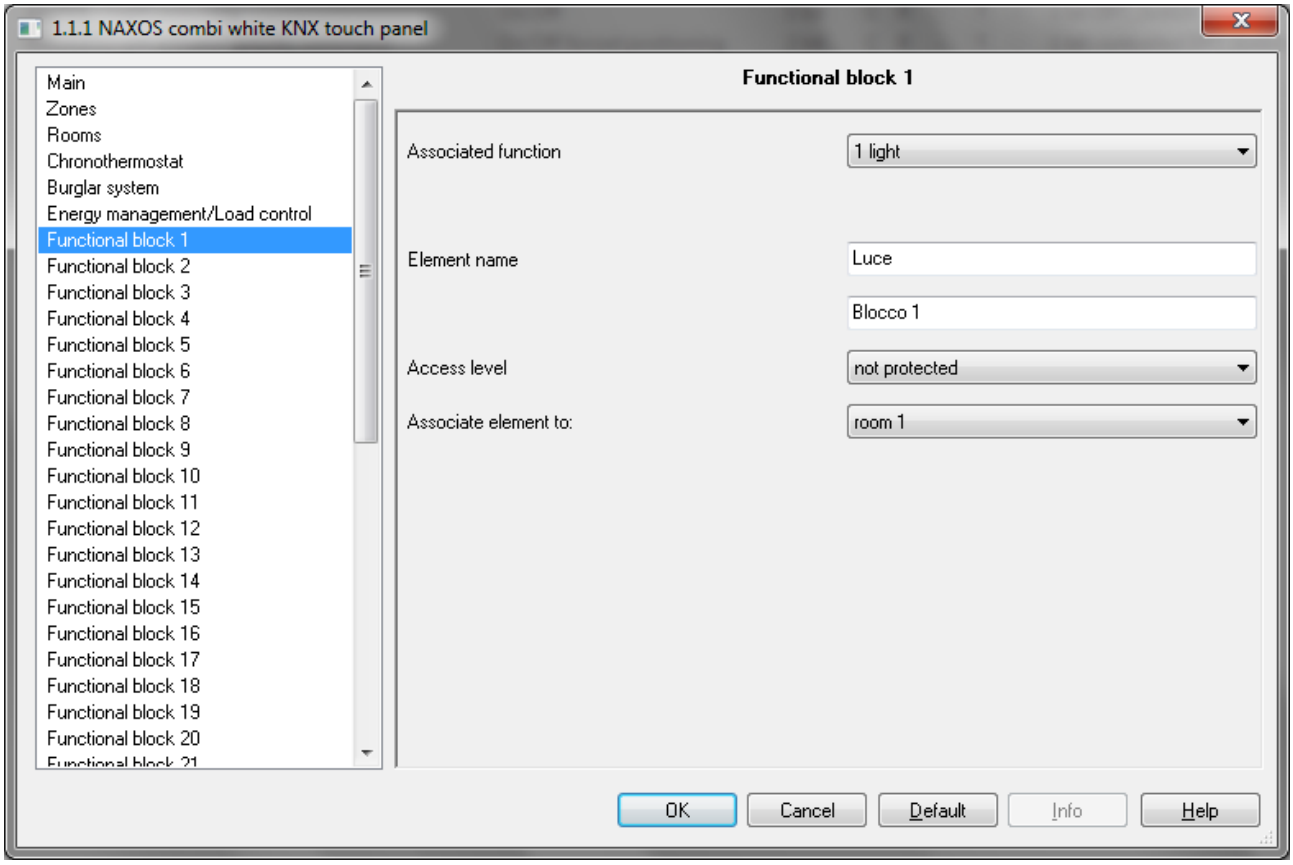


Fig 11: Setting ETS parameters - "1 light function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 1 light element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

6.8 "2 lights" function

This function is used to create two graphic elements for controlling two lights with some of the main control components.
The following command becomes available:

- ON/OFF managed by the object ***Block xA - Light switching*** (Date Point Type: 1.001 DPT_Switch) for the first element and ***Block xB - Light switching*** (Date Point Type: 1.001 DPT_Switch) for the second

In the same way, the graphic element is always able to display:

- the ON or OFF status of the light, received from the BUS via the object ***Block xA - Light status feedback*** (Date Point Type: 1.001 DPT_Switch) for the first element and ***Block xB - Light status feedback*** (Date Point Type: 1.001 DPT_Switch) for the second

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block xA - Light status feedback*** and ***Block xB - Light status feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "Lighting" category.

The basic structure of the menu is as follows:

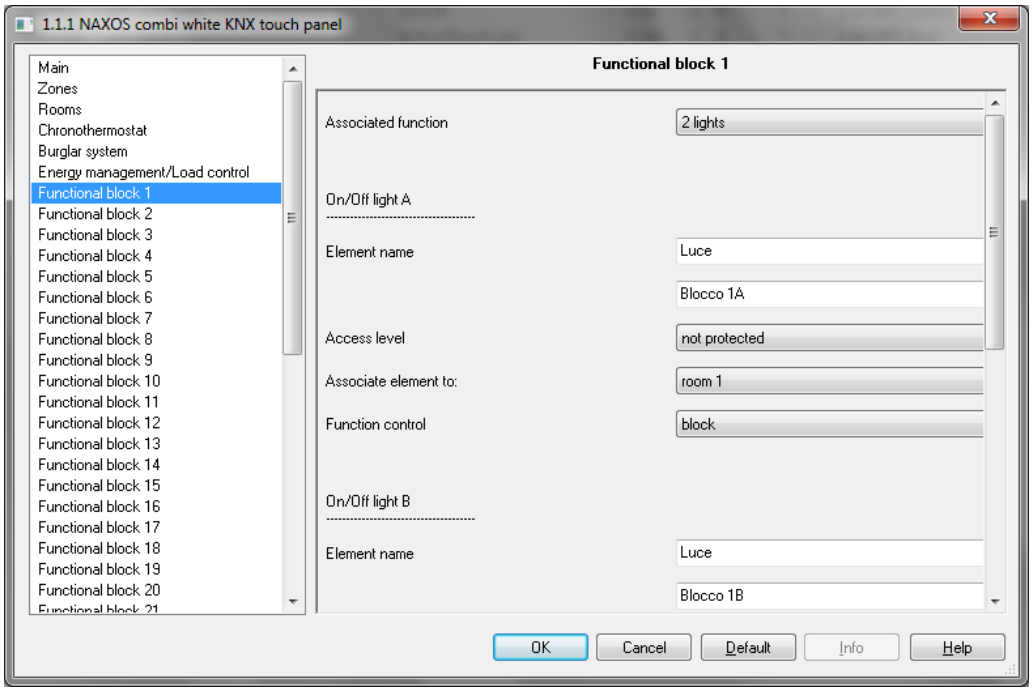


Fig 12: Setting ETS parameters - "2 lights function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 2 light element (A or B, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A or B, depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A or B, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Function control"** is used to expand the range of commands for controlling the 2 light type element (A or B, depending on the header to which the parameter refers); in addition to the command that was already mentioned and always enabled, the 2 light element can be configured to send block on/off commands or priority commands (forcing).
The values that can be set are:

- **block** **(default value)**
- forcing

selecting the value **block** displays the communication object ***Block xA - Light block*** (Date Point Type: 1.002 DPT_Bool) for the first element and ***Block xB - Light block*** (Date Point Type: 1.002 DPT_Bool) for the second; vice versa, selecting the value **forcing** displays the communication object ***Block xA - Light priority command*** (Date Point Type: 2.001 DPT_Switch_Control) for the first element and ***Block xB - Light priority command*** (Date Point Type: 2.001 DPT_Switch_Control) for the second.

6.9 "3 lights" function

This function is used to create three graphic elements for controlling three lights with some of the main control components.
The following command becomes available:

- ON/OFF managed by the object ***Block xA - Light switching*** (Date Point Type: 1.001 DPT_Switch) for the first element, ***Block xB - Light switching*** (Date Point Type: 1.001 DPT_Switch) for the second and ***Block xC - Light switching*** (Date Point Type: 1.001 DPT_Switch) for the third

In the same way, the graphic element is always able to display:

- the ON or OFF status of the light, received from the BUS via the object ***Block xA - Light status feedback*** (Date Point Type: 1.001 DPT_Switch) for the first element, ***Block xB - Light status feedback*** (Date Point Type: 1.001 DPT_Switch) for the second and ***Block xC - Light status feedback*** (Date Point Type: 1.001 DPT_Switch) for the third

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block xA - Light status feedback***, ***Block xB - Light status feedback*** and ***Block xC - Light status feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the “Lighting” category.
The basic structure of the menu is as follows:

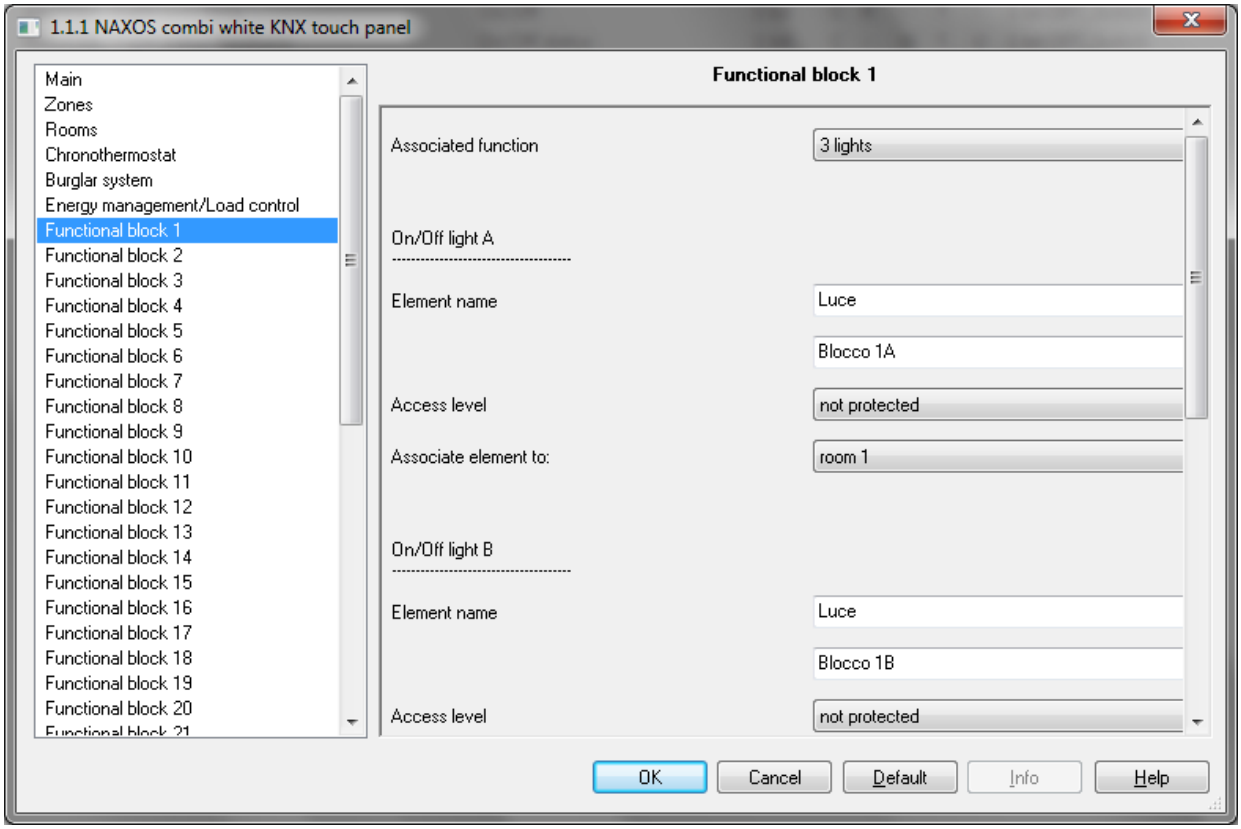


Fig 13: Setting ETS parameters - “3 lights function” section

The parameter **“Element name”** is used to enter the name you want to associate with the 3 light element (A, B or C, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **“Access level”** it is possible to define the rights required to access the commands made available by the element (A, B or C depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A, B or C, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **“Associate element to”**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

6.10 "1 on/off actuator" function

This function is used to create a graphic element for controlling a relay actuator with all the main control components.
The following commands are always available:

- ON/OFF managed by the object ***Block x - On/Off actuator switching*** (Date Point Type: 1.001 DPT_Switch)
- on/off forcing and forcing deactivation (priority commands) managed via the object ***Block x - On/Off actuator priority command*** (Date Point Type: 2.001 DPT_Switch_Control)
- block function on/off managed via the object ***Block x - On/Off actuator block*** (Date Point Type: 1.002 DPT_Bool)

In the same way, the graphic element is always able to display:

- the ON or OFF status of the relay, received from the BUS via the object ***Block x - On/Off actuator status feedback*** (Date Point Type: 1.001 DPT_Switch)

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the object ***Block x - On/Off actuator status feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "On/off actuators" category.

The basic structure of the menu is as follows:

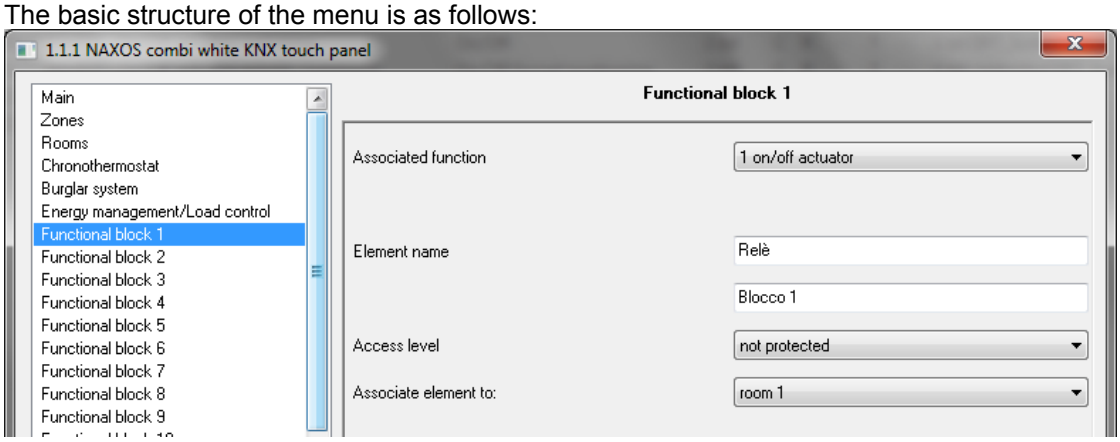


Fig 14: Setting ETS parameters - "1 on/off actuator function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 1 on/off actuator element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

6.11 *"2 on/off actuators"* function

This function is used to create two graphic elements for controlling relay actuators with some of the main control components.
The following command becomes available:

- ON/OFF managed by the object ***Block xA - On/Off actuator switching*** (Date Point Type: 1.001 DPT_Switch) for the first element and ***Block xB - On/Off actuator switching*** (Date Point Type: 1.001 DPT_Switch) for the second

In the same way, the graphic element is always able to display:

- the ON or OFF status of the relay, received from the BUS via the object ***Block xA - On/Off actuator status feedback*** (Date Point Type: 1.001 DPT_Switch) for the first element and ***Block xB - On/Off actuator status feedback*** (Date Point Type: 1.001 DPT_Switch) for the second

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block xA - On/Off actuator status feedback*** and ***Block xB - On/Off actuator status feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "On/off actuators" category.

The basic structure of the menu is as follows:

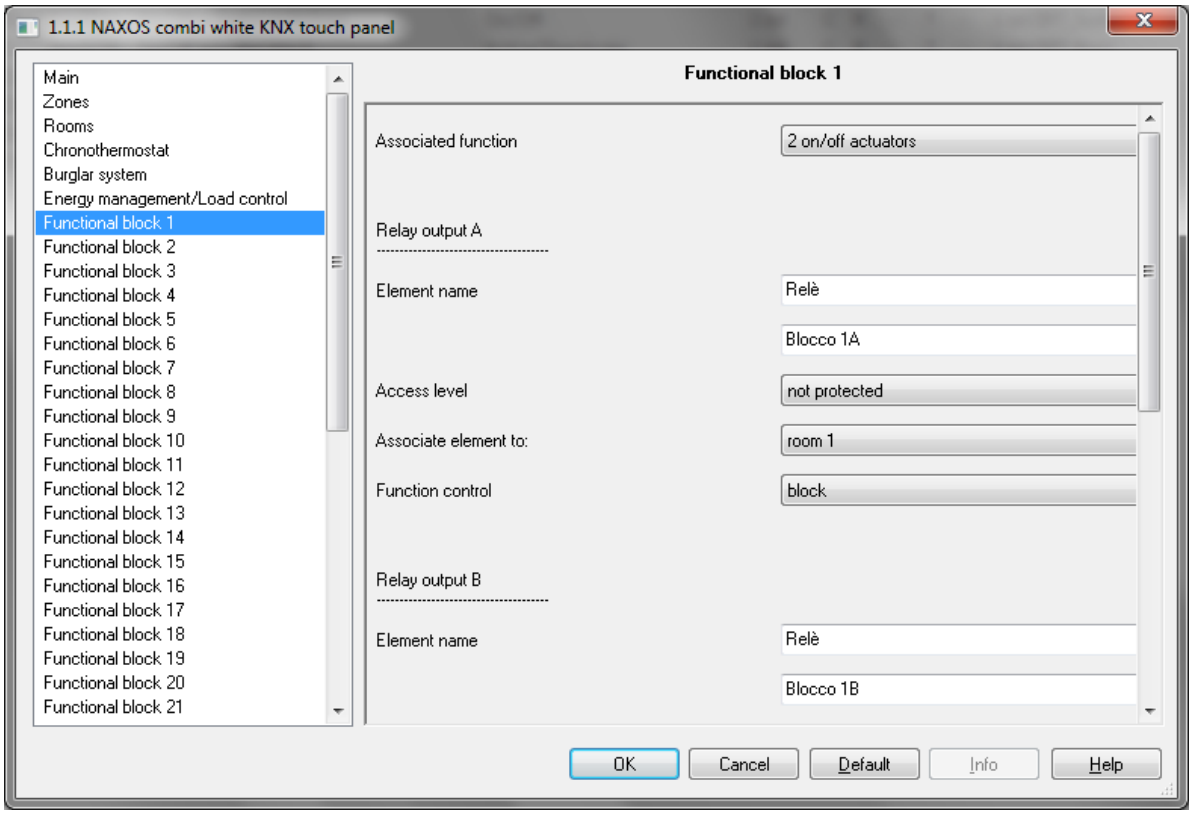


Fig 15: Setting ETS parameters - "2 on/off actuators function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 2 relay output element (A or B, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A or B, depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A or B, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Function control"** is used to expand the range of commands for controlling the 2 output relay type element (A or B, depending on the header to which the parameter refers); in addition to the command that was already mentioned and always enabled, the 2 relay output element can be configured to send block on/off commands or priority commands (forcing).
The values that can be set are:

- **block** **(default value)**
- forcing

selecting the value **block** displays the communication object ***Block xA - On/Off actuator block*** (Date Point Type: 1.002 DPT_Bool) for the first element and ***Block xB - On/Off actuator block*** (Date Point Type: 1.002 DPT_Bool) for the second; vice versa, selecting the value **forcing** displays the communication object ***Block xA - On/Off actuator priority command*** (Date Point Type: 2.001 DPT_Switch_Control) for the first element and ***Block xB - On/Off actuator priority command*** (Date Point Type: 2.001 DPT_Switch_Control) for the second.

6.12 "3 on/off actuators" function

This function is used to create three graphic elements for controlling three relay actuators with some of the main control components.
The following command becomes available:

- ON/OFF managed by the object ***Block xA - On/Off actuator switching*** (Date Point Type: 1.001 DPT_Switch) for the first element, ***Block xB - On/Off actuator switching*** (Date Point Type: 1.001 DPT_Switch) for the second and ***Block xC - On/Off actuator switching*** (Date Point Type: 1.001 DPT_Switch) for the third

In the same way, the graphic element is always able to display:

- the ON or OFF status of the relay, received from the BUS via the object ***Block xA - On/Off actuator status feedback*** (Date Point Type: 1.001 DPT_Switch) for the first element, ***Block xB - On/Off actuator status feedback*** (Date Point Type: 1.001 DPT_Switch) for the second and ***Block xC - On/Off actuator status feedback*** (Date Point Type: 1.001 DPT_Switch) for the third

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block xA - On/Off actuator status feedback***, ***Block xB - On/Off actuator status feedback*** and ***Block xC - On/Off actuator status feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "On/off actuators" category.

The basic structure of the menu is as follows:

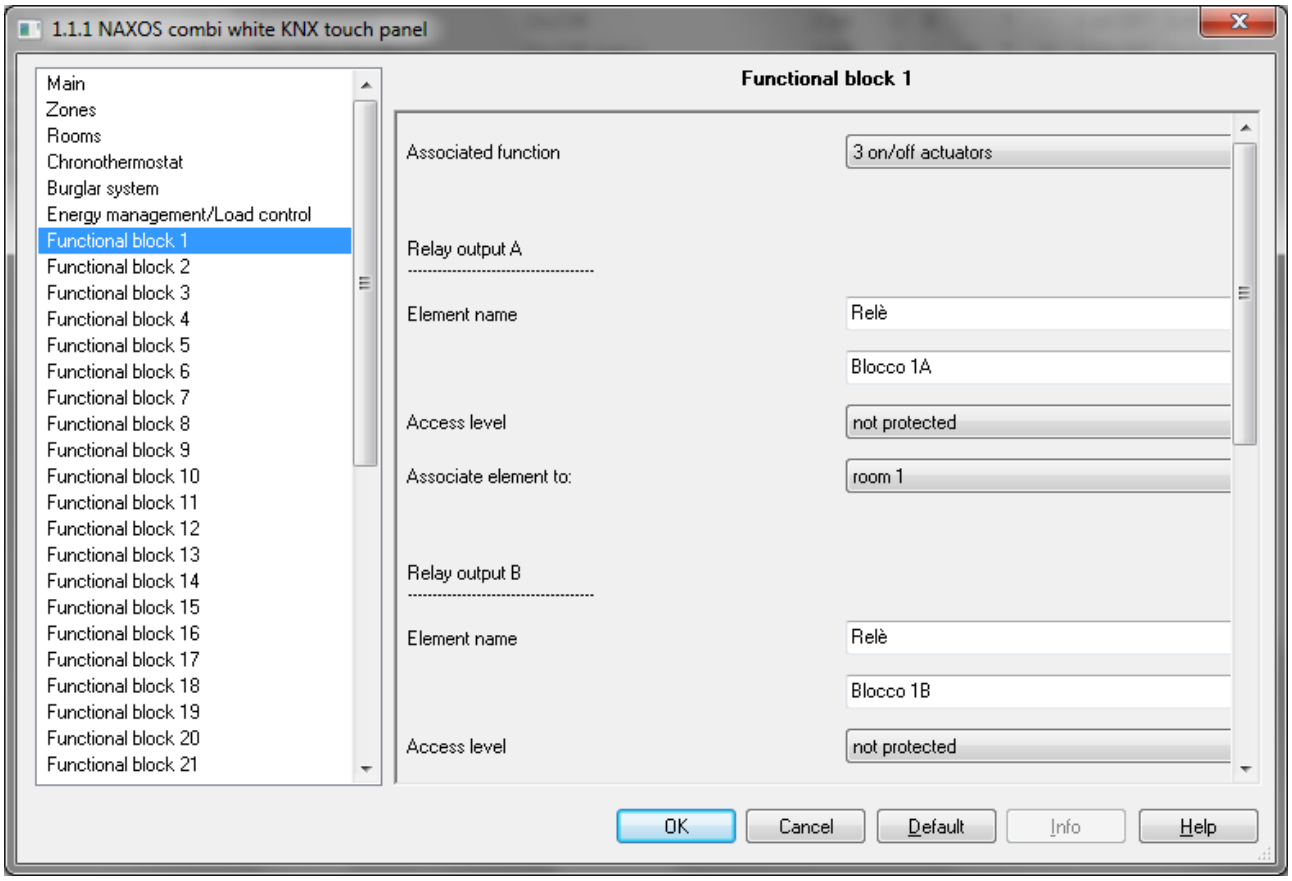


Fig 16: Setting ETS parameters - "3 on/off actuators function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 3 relay output element (A, B or C, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A, B or C depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A, B or C, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

6.13 "1 shutter" function

This function is used to create a graphic element for controlling a shutter or venetian blind with all the main control components.

The graphic element that represents this function, inside the navigation tree belongs to the "Shutters" category.

The basic structure of the menu is as follows:

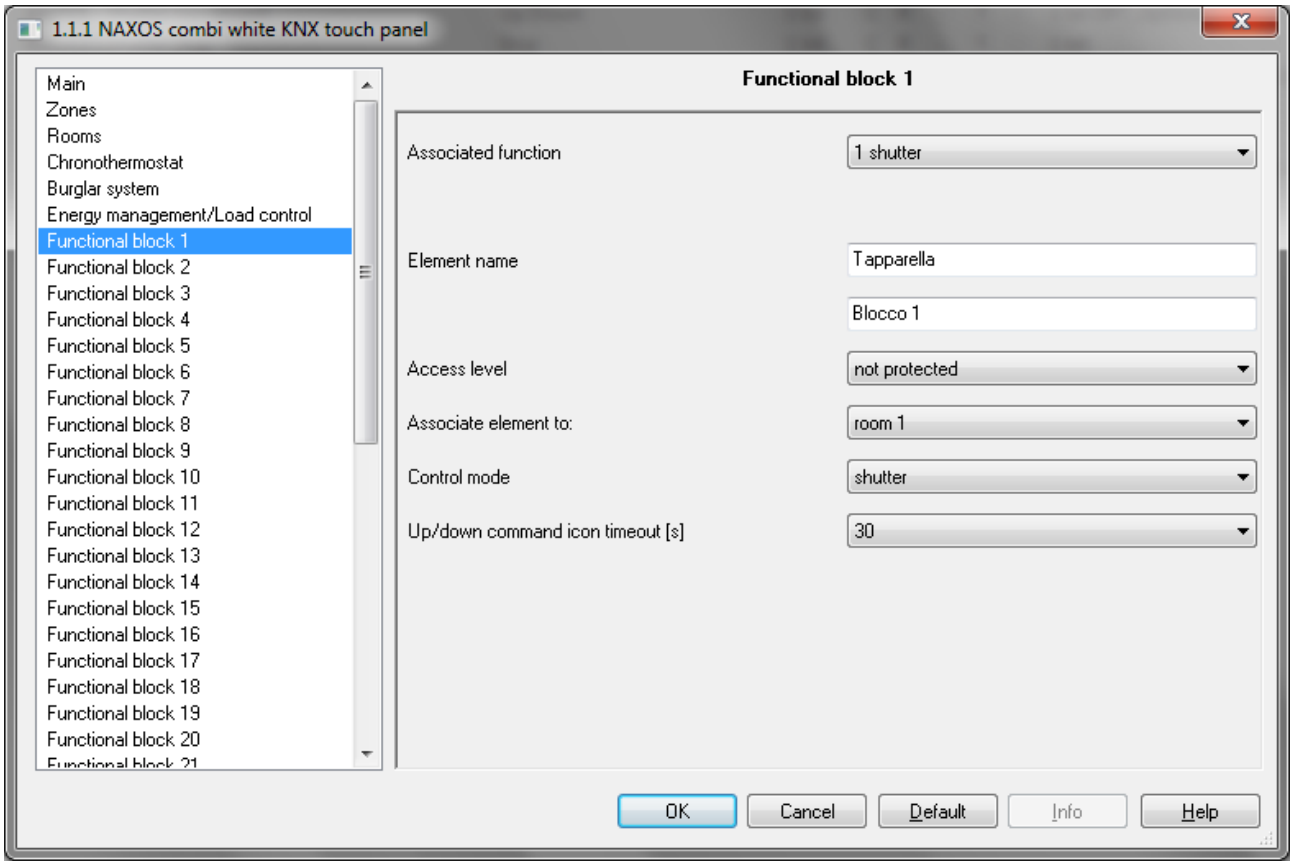


Fig 17: Setting ETS parameters - "1 shutter function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 1 shutter element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3

- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Control mode"** is used to define the type of load you want to control, so that also the graphic element used by the panel is appropriate for that certain operation. The values that can be set are:

- **shutter (default value)**
  Selecting this value, the graphic element makes the following commands available:
  - up/down movement managed via the object ***Block x - Shutter movement*** (Date Point Type: 1.008 DPT_UpDown)
  - stop current movement managed via the object ***Block x - Shutter stop*** (Date Point Type: 1.017 DPT_Trigger)
  - absolute percentage position setting managed via the object ***Block x - Shutter position command*** (Date Point Type: 5.001 DPT_Scaling)
  - up/down forcing and forcing deactivation (priority commands) managed via the object ***Block x - Shutter priority command*** (Date Point Type: 2.008 DPT_Direction1_Control)
  - block function on/off managed via the object ***Block x - Shutter block*** (Date Point Type: 1.002 DPT_Bool)

  In the same way, the graphic element is able to display:
  - the current percentage position of the shutter, received from the BUS via the object ***Block x - Shutter position feedback*** (Date Point Type: 5.001 DPT_Scaling)

  When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the object ***Block x - Shutter position feedback*** to update the status information.

- venetian blind
  Selecting this value, the graphic element makes the following commands available:
  - up/down movement managed via the object ***Block x - Venetian movement*** (Date Point Type: 1.008 DPT_UpDown)
  - stop current movement and adjustment of opening/closing louvres managed via the object ***Block x - Venetian stop/louvres control*** (Date Point Type: 1.007 DPT_Step)
  - absolute percentage position setting managed via the object ***Block x - Venetian position command*** (Date Point Type: 5.001 DPT_Scaling)
  - up/down forcing and forcing deactivation (priority commands) managed via the object ***Block x - Venetian priority command*** (Date Point Type: 2.008 DPT_Direction1_Control)
  - block function on/off managed via the object ***Block x - Venetian block*** (Date Point Type: 1.002 DPT_Bool)

  In the same way, the graphic element is able to display:
  - the current percentage position of the venetian blind, received from the BUS via the object ***Block x - Venetian position feedback*** (Date Point Type: 5.001 DPT_Scaling)

  When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the object ***Block x - Venetian position feedback*** to update the status information.

The parameter **"Up/down command icon timeout [s]"** is used to define the period that the shutter/venetian blind takes to complete a complete movement; this value is particularly useful to the graphic element that manages the feedback of load movement in progress (information that cannot be obtained directly from the shutter/venetian blind via the BUS) and, as a result, sends the movement stop commands; if the venetian blind is operating, the stop command is only sent if there is movement in progress (command icon timeout not expired) to avoid creating a misalignment between the graphic element and the reality, as both the venetian blind stop commands as well as the louvre regulation commands are sent via the same object (***Block x - Venetian stop/louvres control***).
The values that can be set are:

- from 10 to 255 with steps of 1

6.14 "2 shutters" function

This function is used to create two graphic elements for controlling two shutters or venetian blinds with some of the main control components.

The graphic element that represents this function, inside the navigation tree belongs to the "Shutters" category.

The basic structure of the menu is as follows:

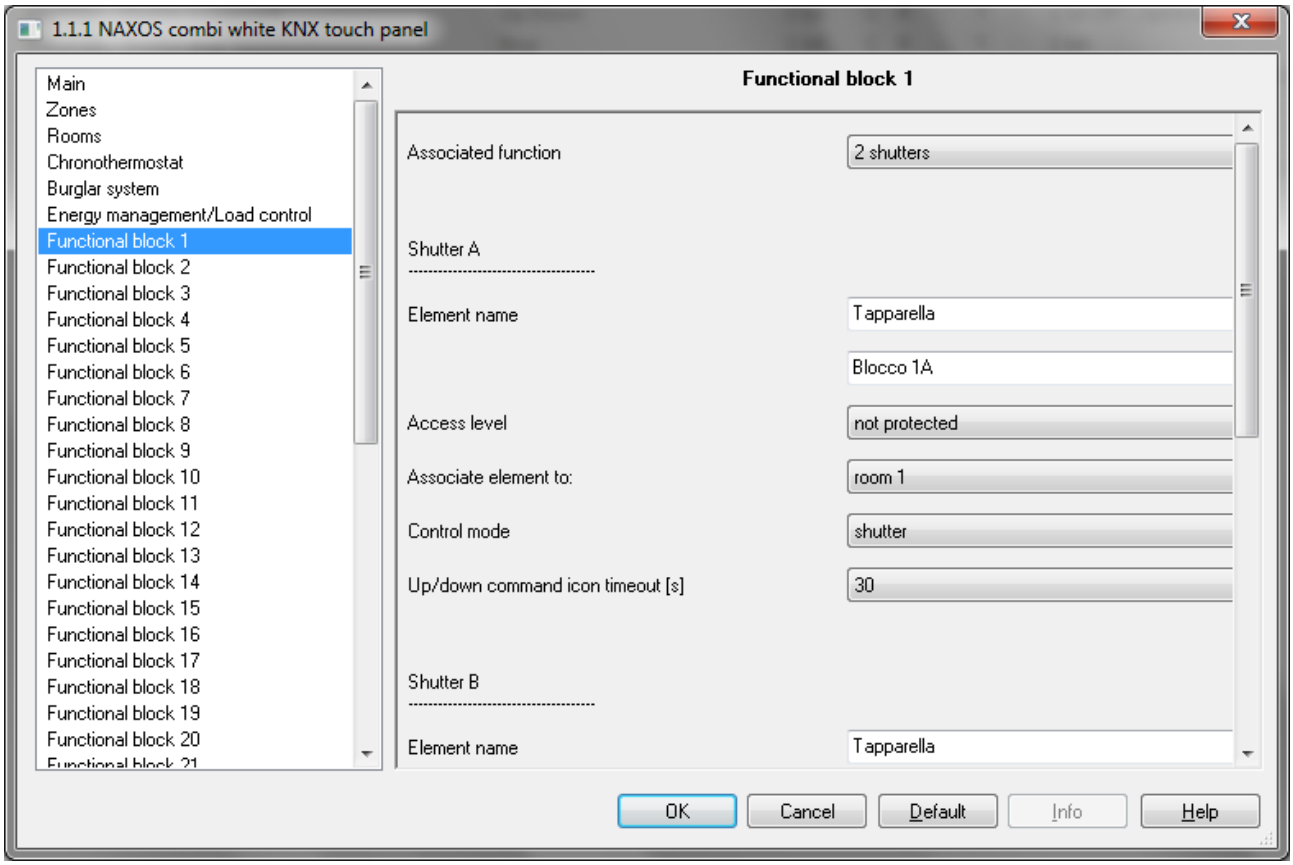


Fig 18: Setting ETS parameters - "3 shutters function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 2 shutter element (A or B, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A or B, depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A or B, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2

- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Control mode"** is used to define the type of load you want to control, so that also the graphic element (A or B, depending on the header to which the parameter refers) used by the panel is appropriate for that certain operation. The values that can be set are:

- **shutter (default value)**
  Selecting this value, the graphic element makes the following commands available:
  - up/down movement managed via the object ***Block xA - Shutter movement*** (Date Point Type: 1.008 DPT_UpDown) for the first element and ***Block xB - Shutter movement*** (Date Point Type: 1.008 DPT_UpDown) for the second
  - stop current movement managed via the object ***Block xA - Shutter stop*** (Date Point Type: 1.017 DPT_Trigger) for the first element and ***Block xB - Shutter stop*** (Date Point Type: 1.017 DPT_Trigger) for the second

  In the same way, the graphic element is able to display:
  - the current percentage position of the shutter, received from the BUS via the object ***Block xA - Shutter position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the first element and ***Block xB - Shutter position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the second

  When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block xA - Shutter position feedback*** and ***Block xB - Shutter position feedback*** to update the status information.

- venetian blind
  Selecting this value, the graphic element makes the following commands available:
  - up/down movement managed via the object ***Block xA - Venetian movement*** (Date Point Type: 1.008 DPT_UpDown) for the first element and ***Block xB - Venetian movement*** (Date Point Type: 1.008 DPT_UpDown) for the second
  - stop current movement and adjustment of opening/closing louvres managed via the object ***Block xA - Venetian stop/louvres control*** (Date Point Type: 1.007 DPT_Step) for the first element and ***Block xB - Venetian stop/louvres control*** (Date Point Type: 1.007 DPT_Step) for the second

  In the same way, the graphic element is able to display:
  - the current percentage position of the venetian blind, received from the BUS via the object ***Block xA - Venetian position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the first element and ***Block xB - Venetian position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the second

  When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block xA - Venetian position feedback*** and ***Block xB - Venetian position feedback*** to update the status information.

The parameter **"Up/down command icon timeout [s]"** is used to define the period that the shutter/venetian blind takes to complete a complete movement; this value is particularly useful to the graphic element (A or B, depending on the header to which the parameter refers) that manages the feedback of load movement in progress (information that cannot be obtained directly from the shutter/venetian blind via the BUS) and, as a result, sends the movement stop commands; if the venetian blind is operating, the stop command is only sent if there is movement in progress (command icon timeout not expired) to avoid creating a misalignment between the graphic element and the reality, as both the venetian blind stop commands as well as the louvre regulation commands are sent via the same object (***Block xA - Venetian stop/louvres control*** or ***Block xB - Venetian stop/louvres control***).
The values that can be set are:

- from 10 to 255 with steps of 1,

6.15 "3 shutters" function

This function is used to create three graphic elements for controlling three shutters or venetian blinds with some of the main control components.

The graphic element that represents this function, inside the navigation tree belongs to the "Shutters" category.

The basic structure of the menu is as follows:

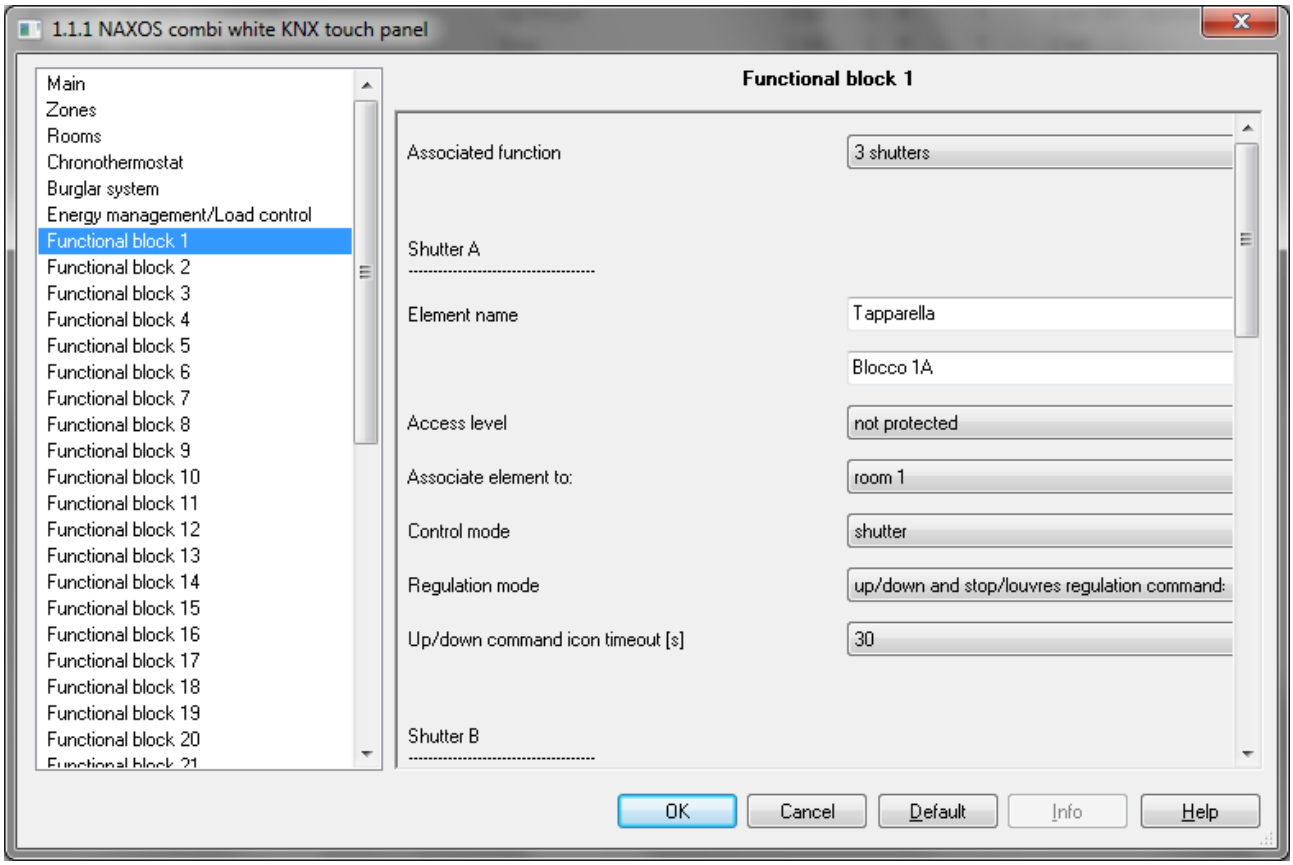


Fig 19: Setting ETS parameters - "3 shutters function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 3 shutters element (A, B or C, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A, B or C depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element (A, B or C, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3

- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Control mode"** is used to define the type of load you want to control, so that also the graphic element (A, B or C, depending on the header to which the parameter refers) used by the panel is appropriate for that certain operation. The values that can be set are:

- **shutter (default value)**
- venetian blind

The parameter **"Regulation mode"** is used to select the type of commands that the graphic element (A, B or C depending on the header to which the parameter refers) makes available. The values that can be set are:

- **up/down and stop/louvres regulation commands (default value)**
  Selecting this value, if the control mode is "shutters", the graphic element makes the following commands available:
  - up/down movement managed via the object ***Block xA - Shutter movement*** (Date Point Type: 1.008 DPT_UpDown) for the first element, ***Block xB - Shutter movement*** (Date Point Type: 1.008 DPT_UpDown) for the second and ***Block xC - Shutter movement*** (Date Point Type: 1.008 DPT_UpDown) for the third
  - stop current movement managed via the object ***Block xA - Shutter stop*** (Date Point Type: 1.017 DPT_Trigger) for the first element, ***Block xB - Shutter stop*** (Date Point Type: 1.017 DPT_Trigger) for the second and ***Block xC - Shutter stop*** (Date Point Type: 1.017 DPT_Trigger)

  Selecting this value, if the control mode is "venetian blind", the graphic element makes the following commands available:
  - up/down movement managed via the object ***Block xA - Venetian movement*** (Date Point Type: 1.008 DPT_UpDown) for the first element, ***Block xB - Venetian movement*** (Date Point Type: 1.008 DPT_UpDown) for the second and ***Block xC - Venetian movement*** (Date Point Type: 1.008 DPT_UpDown) for the third
  - stop current movement and adjustment of opening/closing louvres managed via the object ***Block xA - Venetian stop/louvres control*** (Date Point Type: 1.007 DPT_Step) for the first element, ***Block xB - Venetian stop/louvres control*** (Date Point Type: 1.007 DPT_Step) for the second and ***Block xC - Venetian stop/louvres control*** (Date Point Type: 1.007 DPT_Step) for the third

  In both cases, objects for the display of the shutter/venetian blind percentage position are not available.

- percentage position command and feedback
  Selecting this value, if the control mode is "shutters", graphic element makes the following available:
  - absolute percentage position setting commands managed via the object ***Block xA - Shutter position command*** (Date Point Type: 5.001 DPT_Scaling) for the first element, ***Block xB - Shutter position command*** (Date Point Type: 5.001 DPT_Scaling) for the second and ***Block xC - Shutter position command*** (Date Point Type: 5.001 DPT_Scaling) for the third
  - the feedback of the current percentage position of the shutter, received from the BUS via the object ***Block xA - Shutter position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the first element, ***Block xB - Shutter position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the second and ***Block xC - Shutter position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the third. When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via these objects to update the status information.

  Selecting this value, if the control mode is "venetian blind", graphic element makes the following available:
  - absolute percentage position setting commands managed via the object ***Block xA - Venetian position command*** (Date Point Type: 5.001 DPT_Scaling) for the first element, ***Block xB - Venetian position command*** (Date Point Type: 5.001 DPT_Scaling) for the second and ***Block xC - Venetian position command*** (Date Point Type: 5.001 DPT_Scaling) for the third
  - the feedback of the current percentage position of the venetian blind, received from the BUS via the object ***Block xA - Venetian position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the first element, ***Block xB - Venetian position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the second and ***Block xC - Venetian position feedback*** (Date Point Type: 5.001 DPT_Scaling) for the

third. When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via these objects to update the status information.

The parameter **"Up/down command icon timeout [s]"**, which is visible if the regulation mode is "louvre up/down and stop/regulation commands", makes it possible to define the period that the shutter/venetian blind takes to complete a complete movement; this value is particularly useful to the graphic element (A, B or C, depending on the header to which the parameter refers) that manages the feedback of load movement in progress (information that cannot be obtained directly from the shutter/venetian blind via the BUS) and, as a result, sends the movement stop commands; if the venetian blind is operating, the stop command is only sent if there is movement in progress (command icon timeout not expired) to avoid creating misalignment between the graphic element and the reality, as both the venetian blind stop commands as well as the louvre regulation commands are sent via the same object (***Block xA - Venetian stop/louvres control***, ***Block xB - Venetian stop/louvres control*** or ***Block xC - Venetian stop/louvres control***).
The values that can be set are:
- from 10 to 255 with steps of 1

6.16 "HVAC master" function

This function is used to create a graphic element for the remote control of a temperature adjustment control device (chronothermostat, thermostat or probe) with all the main control components.
The following commands are always available:
- setting the operating type (heating/cooling) managed via the object ***Block x - Functioning type sending*** (Date Point Type: 1.100 DPT_Heat/Cool)

In the same way, the graphic element is always able to display:
- the type of operation (heating/cooling) currently set, received from the BUS via the object ***Block x - Functioning type feedback*** (Date Point Type: 1.100 DPT_Heat/Cool)
- the temperature currently measured, received from the BUS via the object ***Block x - Measured temperature feedback***
- the operating setpoint currently set, received from the BUS via the object ***Block x - Setpoint feedback***

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects ***Block x - Functioning type feedback***, ***Block x - Measured temperature feedback*** and ***Block x - Setpoint feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the "Temperature adjustment" category.
The basic structure of the menu is as follows:

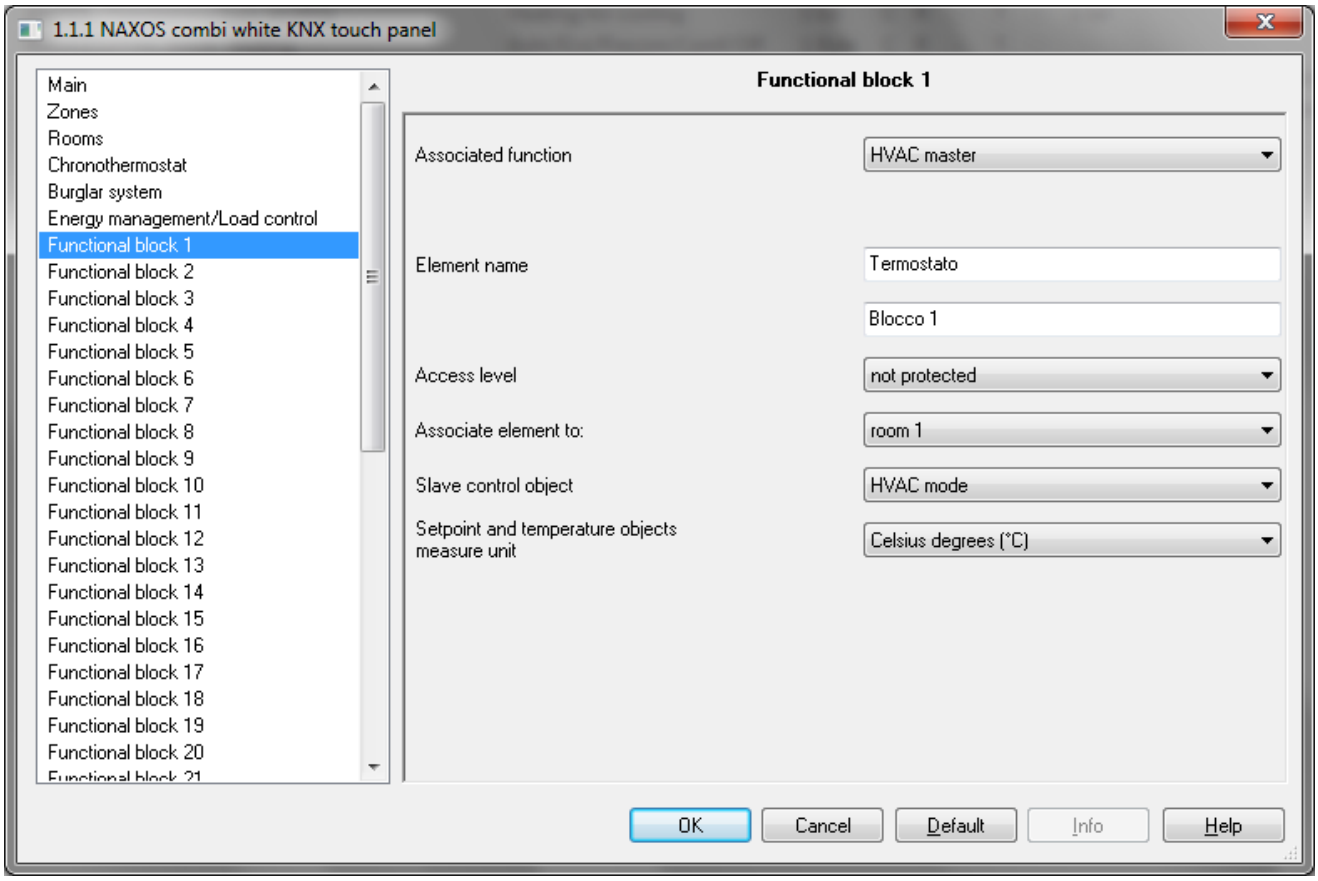


Fig 20: Setting ETS parameters - "HVAC master function" section

The parameter **"Element name"** is used to enter the name you want to associate with the HVAC master element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

If the system in which the panel is installed is equipped with one or more devices for controlling temperature adjustment, the panel can be used as an operating parameter display and setting unit (Master), delegating valve control to the other elements (Slave).
The parameter **"Slave control object"** is used to define if the panel controls the associated remote device through a single setpoint or by changing the active HVAC mode. The values that can be set are:

- **HVAC mode** **(default value)**

The device controls the slave devices, giving them the HVAC operating modes to which they must adjust; in this case, each time the operating mode is selected in the local navigation menu, the value is forwarded immediately to the slave devices. If the selected operating mode is "ACTIVATE PROFILE", the various temperature adjustment modes are sent to the devices, depending on the time profile; this means that at every moment in which a temperature adjustment mode change is set in the time profile, the device sends the functioning mode active in the profile to the slave devices. This displays the communication objects ***Block x - HVAC mode sending*** (Date Point Type: 20.102 DPT_HVACMode) and ***Block x - HVAC mode feedback*** (Date Point Type: 20.102 DPT_HVACMode), used respectively for sending the HVAC mode to the slave device and to receive feedback about the active HVAC mode to be displayed. When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the object ***Block x - HVAC mode feedback*** to update the status information.

- setpoint
  The device controls the slave devices, giving them the operating setpoint value to which they must adjust; in this case, each time the operating setpoint value is selected in the local navigation menu, the value is forwarded immediately to the slave devices. If the selected value is "ACTIVATE PROFILE", the various setpoints are sent to the devices depending on the time profile; this means that at every moment in which a setpoint change is set in the time profile, the device sends the operating setpoint active in the profile to the slave devices. This displays the communication object ***Block x - Setpoint sending***, used for sending the operating setpoint value to the slave.

The parameter **"Setpoint and temperature objects measure unit"** is used to define the unit of measure and as a result the format of the communication objects ***Block x - Measured temperature feedback***, ***Block x - Setpoint feedback*** and ***Block x - Setpoint sending***. The values that can be set are:
- **Celsius degrees (°C)** **(default value)**
- Kelvin degrees (°K)
- Fahrenheit degrees (°F)

The value set for this parameter changes the coding of the communication object indicated above: *9.001 DPT_Value_Temp* if the value is **Celsius degrees (°C)**, *9.002 DPT_Value_Tempd* if the value is **Kelvin degrees (°K)** and *9.027 DPT_Value_Temp_F* if the value is **Fahrenheit degrees (°F)**.

6.17 "Irrigation" function

This function is used to create two graphic elements for the remote control of two irrigators, with all of the main control components. These elements will then be used within an irrigation cycle to implement the "Irrigation" function.
The following commands are always available:
- irrigator on/off managed by the object ***Block xA - Irrigator switching*** (Date Point Type: 1.001 DPT_Switch) for the first element and ***Block xB - Irrigator switching*** (Date Point Type: 1.001 DPT_Switch) for the second

In the same way, the graphic element always is able to:
- display the activation status of an irrigator (on/off), received from the BUS via the object ***Block xA - Irrigator status feedback*** (Date Point Type: 1.001 DPT_Switch) for the first element and ***Block xB - Irrigator status feedback*** (Date Point Type: 1.001 DPT_Switch) for the second
- receive the irrigator block on/off commands received from the BUS via the object ***Block xA - Irrigator block*** (Date Point Type: 1.003 DPT_Enable) for the first element and ***Block xB - Irrigator block*** (Date Point Type: 1.003 DPT_Enable) for the second. These objects are used to temporarily exclude the irrigator from the irrigation cycle, as can take place from the local navigation menu with the specific command.

When the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the object ***Block x - Irrigator status feedback*** to update the status information.

The graphic element that represents this function, inside the navigation tree belongs to the “Functions” category.

The basic structure of the menu is as follows:

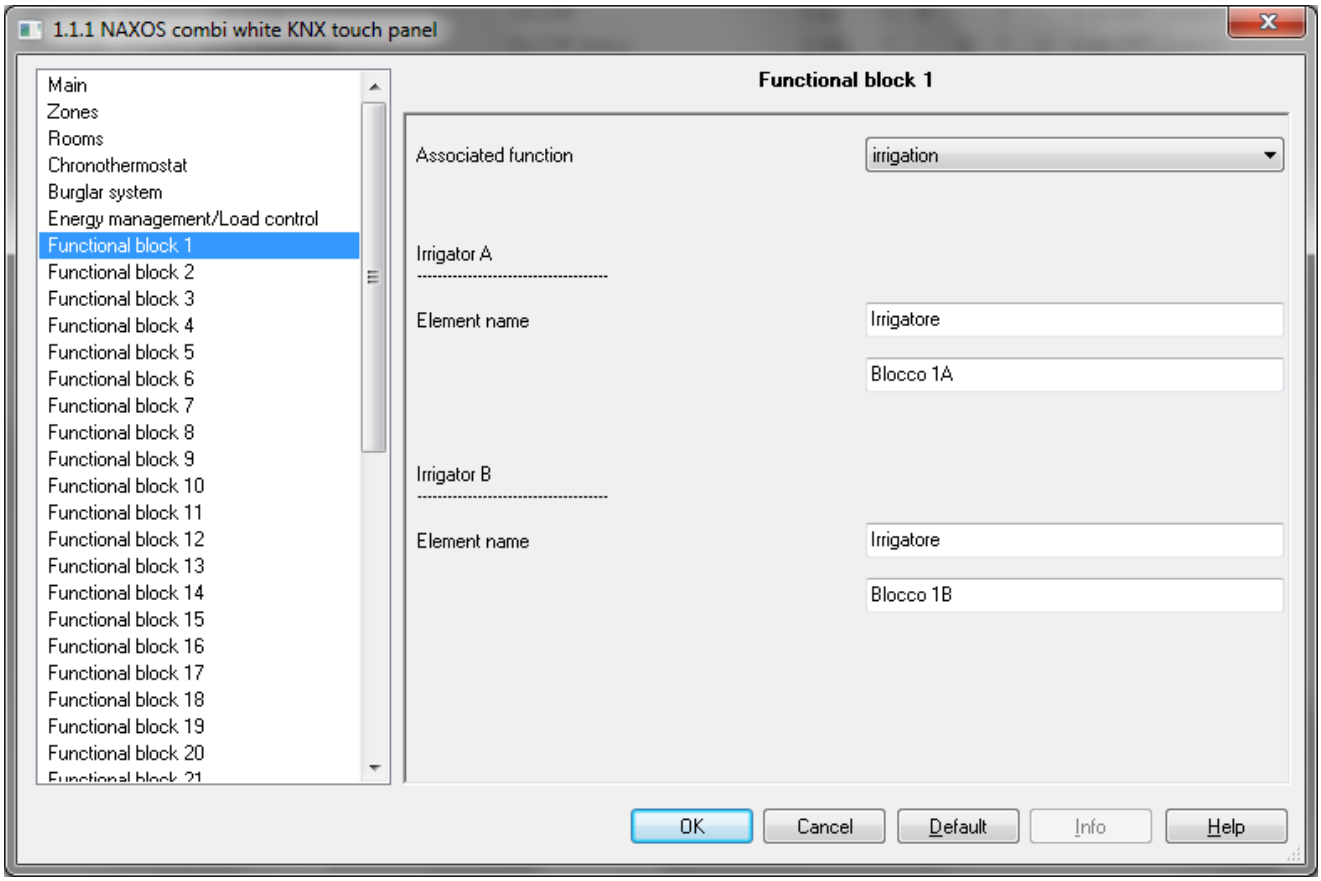


Fig 21: Setting ETS parameters - “Irrigator function" section

The parameter **“Element name”** is used to enter the name you want to associate with the irrigator element (A or B, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

6.18 “6 independent inputs” function

This function is used to create six independent graphic elements to display analogue or digital values; each element is combined with the reciprocal communication object, used to receive the information from the BUS and update the displayed value.

The graphic element that represents this function, inside the navigation tree belongs to the “Inputs” category.

The basic structure of the menu is as follows:

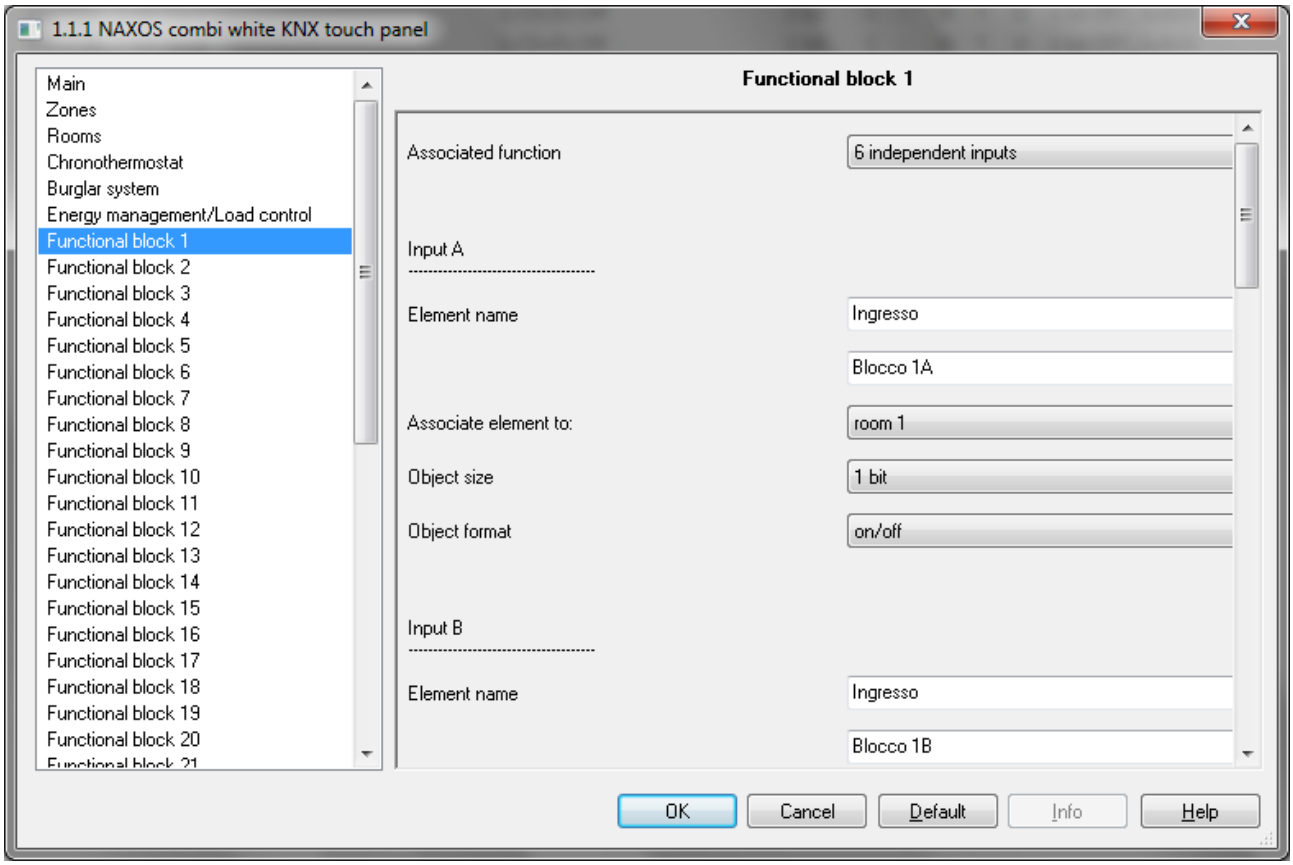


Fig 22: Setting ETS parameters - "6 independent inputs function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 6 independent inputs element (A, B C, D, E or F, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

In the navigation structure, each element (A, B C, D, E or F, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Object size"** is used to set the size of the telegram that will be received by the BUS, whose value will be subsequently displayed by the associated graphic element (A, B, C, D, E or F depending on the header to which the parameter refers). The values that can be set are:

- **1 bit** **(default value)**
- 1 byte
- 2 bytes

- 3 bytes
- 4 bytes

Based on the value set for this item, the values that can be set for the parameter **"Object format"** will change as a result.
The parameter **"Object format"** is used to set the format and the coding of the BUS telegram that will be received by the device via the dedicated communication objects and displayed by the associated graphic element (A, B, C, D, E or F depending on the header to which the parameter refers).

- If the object size is **1 bit**, this displays the communication object ***Block xA - 1 bit input*** for the first element, ***Block xB - 1 bit input*** for the second, ***Block xC - 1 bit input*** for the third, ***Block xD - 1 bit input*** for the fourth, ***Block xE - 1 bit input*** for the fifth and ***Block xF - 1 bit input*** for the sixth; the values that can be set for the above parameter are:

  - **on/off (default value)**
    with this setting, the object format is *1.001 DPT_Switch*
  - boolean
    with this setting, the object format is *1.002 DPT_Bool*
  - heating/cooling temperature adjustment
    with this setting, the object format is *1.100 DPT_Heat/Cool*
  - enable/disable
    with this setting, the object format is *1.003 DPT_Enable*
  - alarm/no alarm
    with this setting, the object format is *1.005 DPT_Alarm*
  - occupied/free
    with this setting, the object format is *1.018 DPT_Occupancy*
  - open/closed
    with this setting, the object format is *1.009 DPT_OpenClose*
  - input with periodic transmission control
    with this setting, the object format is *1.001 DPT_Switch*
  - window contact
    with this setting, the object format is *1.019 DPT_Window/Door*

- If the object size is **1 byte**, this displays the communication object ***Block xA - 1 byte input*** for the first element, ***Block xB - 1 byte input*** for the second, ***Block xC - 1 byte input*** for the third, ***Block xD - 1 byte input*** for the fourth, ***Block xE - 1 byte input*** for the fifth and ***Block xF - 1 byte input*** for the sixth; the values that can be set for the above parameter are:

  - **unsigned** **(default value)**
    with this setting, the object format is *5.010 DPT_Value_1_Ucount*
  - signed
    with this setting, the object format is *6.010 DPT_Value_1_Count*
  - percentage
    with this setting, the object format is *5.001 DPT_Scaling*
  - two's complement percentage
    with this setting, the object format is *6.001 DPT_Percent_V8*
  - HVAC mode
    with this setting, the object format is *20.102 DPT_HVACMode*
  - angular width (degrees)
    with this setting, the object format is *5.003 DPT_Angle*

- If the object size is **2 bytes**, this displays the communication object ***Block xA - 2 bytes input*** for the first element, ***Block xB - 2 bytes input*** for the second, ***Block xC - 2 bytes input*** for the third, ***Block xD - 2 bytes input*** for the fourth, ***Block xE - 2 bytes input*** for the fifth and ***Block xF - 2 bytes input*** for the sixth; the values that can be set for the above parameter are:

  - **unsigned** **(default value)**
    with this setting, the object format is *7.001 DPT_Value_2_Ucount*
  - signed
    with this setting, the object format is *8.001 DPT_Value_2_Count*
  - time in seconds

with this setting, the object format is *7.005 DPT_TimePeriodSec*
- time in minutes
  with this setting, the object format is *7.006 DPT_TimePeriodMin*
- time in hours
  with this setting, the object format is *7.007 DPT_TimePeriodHrs*
- floating temperature °C
  with this setting, the object format is *9.001 DPT_Value_Temp*
- floating temperature °K
  with this setting, the object format is *9.002 DPT_Value_Tempd*
- floating temperature °F
  with this setting, the object format is *9.027 DPT_Value_Temp_F*
- floating brightness
  with this setting, the object format is *9.004 DPT_Value_Lux*
- floating humidity
  with this setting, the object format is *9.007 DPT_Value_Humidity*
- floating solar intensity
  with this setting, the object format is *9.022 DPT_PowerDensity*
- floating voltage
  with this setting, the object format is *9.020 DPT_Value_Volt*
- floating current
  with this setting, the object format is *9.021 DPT_Value_Current*
- floating power
  with this setting, the object format is *9.024 DPT_Power*
- floating wind speed
  con with this setting, the object format is *9.005 DPT_Value_Wsp*

- If the object size is **3 bytes**, this displays the communication object ***Block xA - 3 bytes input*** for the first element, ***Block xB - 3 bytes input*** for the second, ***Block xC - 3 bytes input*** for the third, ***Block xD - 3 bytes input*** for the fourth, ***Block xE - 3 bytes input*** for the fifth and ***Block xF - 3 bytes input*** for the sixth; the values that can be set for the above parameter are:

  - **day/time** **(default value)**
    with this setting, the object format is *10.001 DPT_TimeOfDay*
  - date
    with this setting, the object format is *11.001 DPT_Date*
  - RGB
    with this setting, the object format is *232.600 DPT_Colour_RGB*

- If the object size is **4 bytes**, this displays the communication object ***Block xA - 4 bytes input*** for the first element, ***Block xB - 4 bytes input*** for the second, ***Block xC - 4 bytes input*** for the third, ***Block xD - 4 bytes input*** for the fourth, ***Block xE - 4 bytes input*** for the fifth and ***Block xF - 4 bytes input*** for the sixth; the values that can be set for the above parameter are:

  - **unsigned** **(default value)**
    with this setting, the object format is *12.001 DPT_Value_4_Ucount*
  - signed
    with this setting, the object format is *13.001 DPT_Value_4_Count*
  - signed active energy
    with this setting, the object format is *13.010 DPT_ActiveEnergy*
  - signed reactive energy
    with this setting, the object format is *13.012 DPT_ReactiveEnergy*
  - signed apparent energy
    with this setting, the object format is *13.011 DPT_ApparantEnergy*
  - signed active energy (kWh)
    with this setting, the object format is *13.013 DPT_ActiveEnergy_kWh*
  - signed reactive energy (kVARh)
    with this setting, the object format is *13.015 DPT_ReactiveEnergy_kVARh*
  - signed apparent energy (kVAh)
    with this setting, the object format is *13.014 DPT_ApparantEnergy_kVAh*
  - floating power
    with this setting, the object format is *14.056 DPT_Value_Power*

- floating power factor
  with this setting, the object format is *14.057 DPT_Value_PowerFactor*
- floating speed
  with this setting, the object format is *14.065 DPT_Value_Speed*
- floating volume
  with this setting, the object format is *14.076 DPT_Value_Volume*

Independently of the size and format, when the BUS or auxiliary voltage is restored, the panel sends the status requests (read request) via the objects enabled to update the status information to be displayed.

The parameter **"Monitoring time"**, which is displayed if the format of the object is "1 bit alarm/no alarm" or "1 bit input with periodic transmission", is used to define the monitoring time of BUS telegram reception on the associated object; the values that can be set are:

- from 0 minutes (no monitoring) to 255 minutes with steps of 1, **10 (default value)**

by selecting the value **0 minutes (no monitoring)**, the object enabled for the input will not be monitored.
The meaning of the monitoring time is: if, within the set monitoring time, the telegram with the value "0=no alarm" is not received periodically, the device will still signal the alarm event as if the value "1=alarm" had been received to signal that there may be a transmitting device malfunction.

6.19 *"6 independent outputs"* function

This function is used to create six independent graphic elements to send analogue or digital commands/values; each element is combined with the reciprocal communication object, used to receive the information from the BUS.
The graphic element that represents this function, inside the navigation tree belongs to the "Outputs" category.
The basic structure of the menu is as follows:

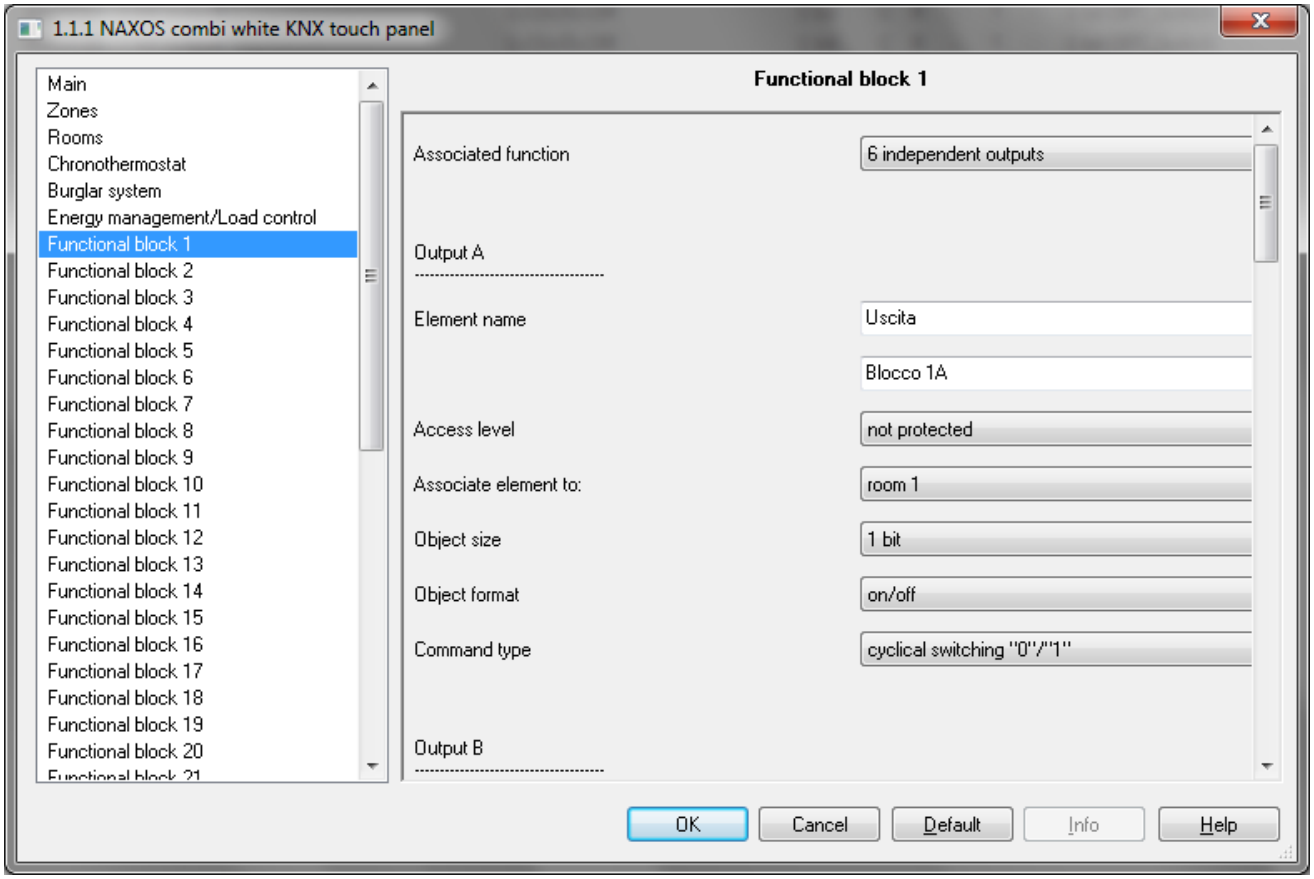


Fig 23: Setting ETS parameters - "6 independent outputs function" section

The parameter **"Element name"** is used to enter the name you want to associate with the 6 independent outputs element (A, B C, D, E or F, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A, B C, D, E or F depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **0 (default value)**
- protection level 1 .. 3

In the navigation structure, each element (A, B C, D, E or F, depending on the header to which the parameter refers) of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32
- not displayed

By selecting **not displayed**, the element will not be present in the Navigation tree.

The parameter **"Object size"** is used to set the size of the telegram that will be sent on the BUS, interacting with the associated graphic element (A, B, C, D, E or F depending on the header to which the parameter refers). The values that can be set are:

- **1 bit (default value)**
- 1 byte
- 2 bytes
- 3 bytes
- 4 bytes

Based on the value set for this item, the values that can be set for the parameter **"Object format"** will change as a result.
The parameter **"Object format"** is used to set the format and the coding of the BUS telegram that will be sent by the panel via the dedicated communication objects interacting with the associated graphic element (A, B, C, D, E or F depending on the header to which the parameter refers).

- If the object size is **1 bit**, this displays the communication object ***Block xA - 1 bit output*** for the first element, ***Block xB - 1 bit output*** for the second, ***Block xC - 1 bit output*** for the third, ***Block xD - 1 bit output*** for the fourth, ***Block xE - 1 bit output*** for the fifth and ***Block xF - 1 bit output*** for the sixth; the values that can be set for the above parameter are:

  - **on/off (default value)**
    with this setting, the object format is *1.001 DPT_Switch*
  - boolean
    with this setting, the object format is *1.002 DPT_Bool*
  - shutters up/down
    with this setting, the object format is *1.008 DPT_UpDown*
  - increase/decrease
    with this setting, the object format is *1.007 DPT_Step*
  - heating/cooling temperature adjustment
    with this setting, the object format is *1.100 DPT_Heat/Cool*
  - enable/disable
    with this setting, the object format is *1.003 DPT_Enable*
  - alarm/no alarm
    with this setting, the object format is *1.005 DPT_Alarm*
  - start/stop
    with this setting, the object format is *1.010 DPT_Start*
  - open/closed
    with this setting, the object format is *1.009 DPT_OpenClose*

- If the object size is **2 bits**, this displays the communication object ***Block xA - 2 bits output*** for the first element, ***Block xB - 2 bits output*** for the second, ***Block xC - 2 bits output*** for the third, ***Block xD - 2 bits output*** for the fourth, ***Block xE - 2 bits output*** for the fifth and ***Block xF - 2 bits output*** for the sixth; the values that can be set for the above parameter are:

  - **on/off forced positioning (default value)**
    with this setting, the object format is *2.001 DPT_Switch_Control*
  - up/down forced positioning
    with this setting, the object format is *2.008 DPT_Direction1_Control*

- If the object size is **4 bits**, this displays the communication object ***Block xA - 4 bits output*** for the first element, ***Block xB - 4 bits output*** for the second, ***Block xC - 4 bits output*** for the third, ***Block xD - 4 bits output*** for the fourth, ***Block xE - 4 bits output*** for the fifth and ***Block xF - 4 bits output*** for the sixth; the values that can be set for the above parameter are:

  - **dimmer step** **(default value)**
    with this setting, the object format is *3.007 DPT_Control_Dimming*
  - shutter step
    with this setting, the object format is *3.008 DPT_Control_Blinds*

- If the object size is **1 byte**, this displays the communication object ***Block xA - 1 byte output*** for the first element, ***Block xB - 1 byte output*** for the second, ***Block xC - 1 byte output*** for the third, ***Block xD - 1 byte output*** for the fourth, ***Block xE - 1 byte output*** for the fifth and ***Block xF - 1 byte output*** for the sixth; the values that can be set for the above parameter are:

  - **unsigned** **(default value)**
    with this setting, the object format is *5.010 DPT_Value_1_Ucount*
  - signed
    with this setting, the object format is *6.010 DPT_Value_1_Count*
  - percentage
    with this setting, the object format is *5.001 DPT_Scaling*
  - two's complement percentage
    with this setting, the object format is *6.001 DPT_Percent_V8*
  - HVAC mode
    with this setting, the object format is *20.102 DPT_HVACMode*
  - angular width (degrees)
    with this setting, the object format is *5.003 DPT_Angle*

- If the object size is **2 bytes**, this displays the communication object ***Block xA - 2 bytes output*** for the first element, ***Block xB - 2 bytes output*** for the second, ***Block xC - 2 bytes output*** for the third, ***Block xD - 2 bytes output*** for the fourth, ***Block xE - 2 bytes output*** for the fifth and ***Block xF - 2 bytes output*** for the sixth; the values that can be set for the above parameter are:

  - **unsigned** **(default value)**
    with this setting, the object format is *7.001 DPT_Value_2_Ucount*
  - signed
    with this setting, the object format is *8.001 DPT_Value_2_Count*
  - time in seconds
    with this setting, the object format is *7.005 DPT_TimePeriodSec*
  - time in minutes
    with this setting, the object format is *7.006 DPT_TimePeriodMin*
  - time in hours
    with this setting, the object format is *7.007 DPT_TimePeriodHrs*
  - floating temperature °C
    with this setting, the object format is *9.001 DPT_Value_Temp*
  - floating temperature °K
    with this setting, the object format is *9.002 DPT_Value_Tempd*
  - floating temperature °F
    with this setting, the object format is *9.027 DPT_Value_Temp_F*
  - floating brightness
    with this setting, the object format is *9.004 DPT_Value_Lux*
  - floating humidity
    with this setting, the object format is *9.007 DPT_Value_Humidity*
  - floating solar intensity
    with this setting, the object format is *9.022 DPT_PowerDensity*
  - floating voltage
    with this setting, the object format is *9.020 DPT_Value_Volt*
  - floating current
    with this setting, the object format is *9.021 DPT_Value_Current*
  - floating power
    with this setting, the object format is *9.024 DPT_Power*

- floating wind speed
  con with this setting, the object format is *9.005 DPT_Value_Wsp*

- If the object size is **3 bytes**, this displays the communication object ***Block xA - 3 bytes output*** for the first element, ***Block xB - 3 bytes output*** for the second, ***Block xC - 3 bytes output*** for the third, ***Block xD - 3 bytes output*** for the fourth, ***Block xE - 3 bytes output*** for the fifth and ***Block xF - 3 bytes output*** for the sixth; the values that can be set for the above parameter are:

  - **day/time (default value)**
    with this setting, the object format is *10.001 DPT_TimeOfDay*
  - date
    with this setting, the object format is *11.001 DPT_Date*
  - RGB
    with this setting, the object format is *232.600 DPT_Colour_RGB*

- If the object size is **4 bytes**, this displays the communication object ***Block xA - 4 bytes output*** for the first element, ***Block xB - 4 bytes output*** for the second, ***Block xC - 4 bytes output*** for the third, ***Block xD - 4 bytes output*** for the fourth, ***Block xE - 4 bytes output*** for the fifth and ***Block xF - 4 bytes output*** for the sixth; the values that can be set for the above parameter are:

  - **unsigned (default value)**
    with this setting, the object format is *12.001 DPT_Value_4_Ucount*
  - signed
    with this setting, the object format is *13.001 DPT_Value_4_Count*
  - signed active energy
    with this setting, the object format is *13.010 DPT_ActiveEnergy*
  - signed reactive energy
    with this setting, the object format is *13.012 DPT_ReactiveEnergy*
  - signed apparent energy
    with this setting, the object format is *13.011 DPT_ApparantEnergy*
  - signed active energy (kWh)
    with this setting, the object format is *13.013 DPT_ActiveEnergy_kWh*
  - signed reactive energy (kVARh)
    with this setting, the object format is *13.015 DPT_ReactiveEnergy_kVARh*
  - signed apparent energy (kVAh)
    with this setting, the object format is *13.014 DPT_ApparantEnergy_kVAh*
  - floating power
    with this setting, the object format is *14.056 DPT_Value_Power*
  - floating power factor
    with this setting, the object format is *14.057 DPT_Value_PowerFactor*
  - floating speed
    with this setting, the object format is *14.065 DPT_Value_Speed*
  - floating volume
    with this setting, the object format is *14.076 DPT_Value_Volume*

The parameter **"Command type**, which is displayed if the size of the object is "1 bit", is used to define the behaviour of the graphic element associated with the object after the element is pressed/released; the values that can be set are:

- send "0" only
- send "1" only
- send "1" when pressed/"0" when released
- send "0" when pressed/"1" when released
- **cyclical switching "0"/"1" (default value)**

The parameter **"Increase/decrease step"**, which is displayed if the format of the object is "4 bits dimmer step" or "4 bits shutter step", is used to set the percentage value of the increase/decrease step associated with the commands sent on the BUS; the values that can be set are:

- **100% (default value)**
- 50%

- 25%
- 12.5%
- 6.25%
- 3.125%
- 1.56%

6.20 "KNX scenes" function

This function is used to create three graphic elements for controlling three independent KNX scenes with all the main control components.

The graphic element that represents this function, inside the navigation tree belongs to the "Functions" category.

The basic structure of the menu is as follows:

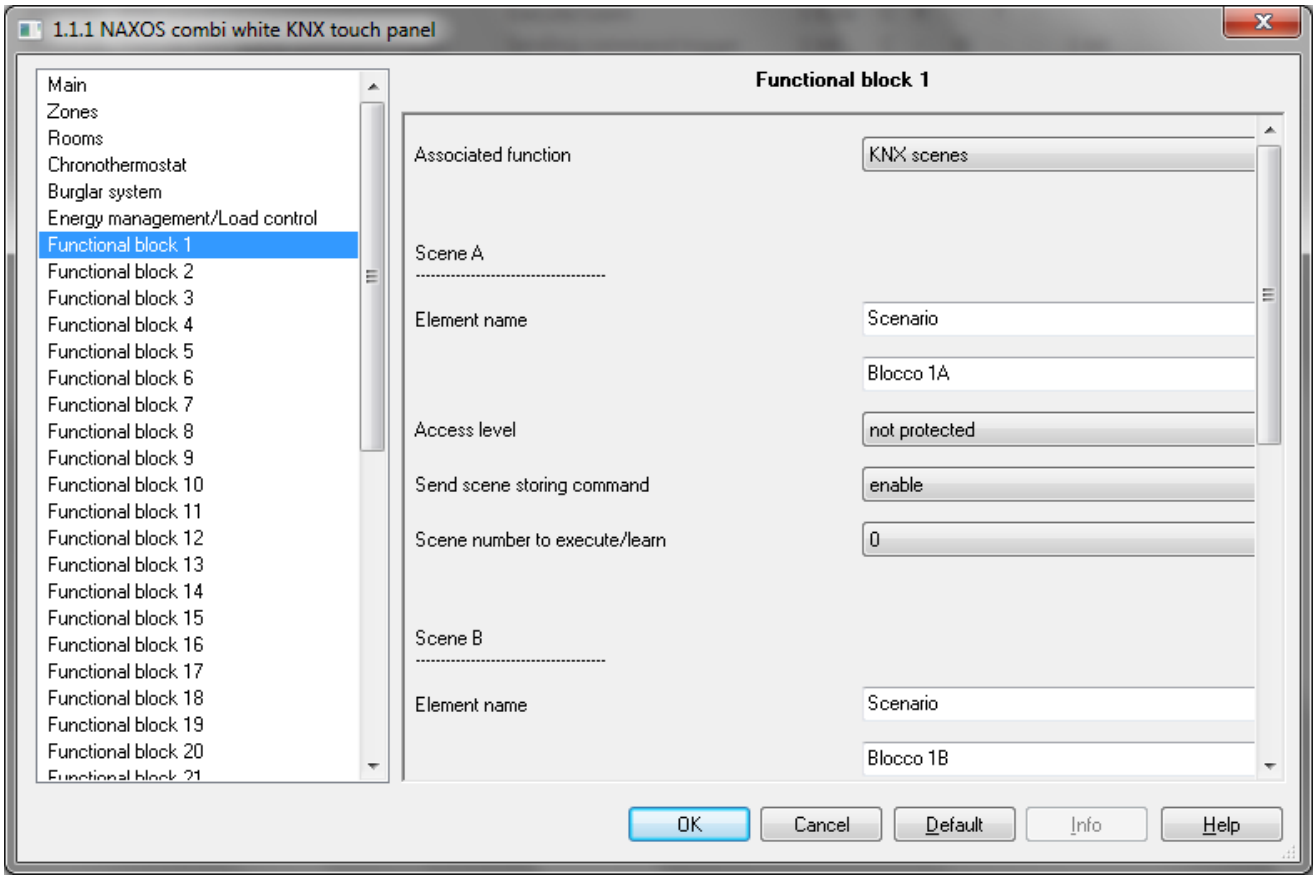


Fig 24:Setting ETS parameters - "KNX scenes function" section

The parameter **"Element name"** is used to enter the name you want to associate with the KNX scene element (A, B or C, depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required to access the commands made available by the element (A, B or C depending on the header to which the parameter refers); even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **0 (default value)**

- protection level 1 .. 3

The parameter **"Send scene storing command"** is used to enable the sending of the scene storing command when recognising a certain interaction with the associated graphic element (A, B or C depending on the header to which the parameter refers). The values that can be set are:

- disable
- **enable (default value)**

only by selecting **enable**, the panel will send the scene storing command after detecting a certain interaction with the associated graphic element (A, B or C depending on the header to which the parameter refers) via the communication object ***Block xA - KNX scene*** (Date Point Type: 18.001 DPT_SceneControl) for the first element, the object ***Block xB - KNX scene*** (Date Point Type: 18.001 DPT_SceneControl) for the second and the object ***Block xC - KNX scene*** (Date Point Type: 18.001 DPT_SceneControl) for the third. Selecting the value **disable**, the storing command is not sent because the interaction with the associated graphic element (A, B or C depending on the header to which the parameter refers) that generates it is not recognised; the scene execution command is sent in its place via the communication object ***Block xA - KNX scene*** (Date Point Type: 17.001 DPT_SceneNumber) for the first element, the object ***Block xB - KNX scene*** (Date Point Type: 17.001 DPT_SceneNumber) for the second and the object ***Block xC - KNX scene*** (Date Point Type: 17.001 DPT_SceneNumber).

The parameter **"Scene number to execute/learn"** is used to set the value of the scene to be executed/learned and as a result, the relative values that are sent by interacting with the associated graphic element (A, B or C depending on the header to which the parameter refers). The possible values are:

- from **0 (default value)** to 63, with steps of 1

In addition to the direct interaction with the graphic element that represents the KNX scene, it is possible to indirectly generate the sending of the scene execution command after receiving a BUS telegram (both with value "1" as well as with value "0") on the object ***Block xA - Execute KNX scene command trigger*** (Date Point Type: 1.017 DPT_Trigger) for the first element, the object ***Block xB - Execute KNX scene command trigger*** (Date Point Type: 1.017 DPT_Trigger) for the second and the object ***Block xC - Execute KNX scene command trigger*** (Date Point Type: 1.017 DPT_Trigger) for the third; each time the panel receives a telegram on this object, it immediately sends the scene execution telegram via the associated objects (***Block xA - KNX scene*** for the first element, the object ***Block xB - KNX scene*** for the second and the object ***Block xC - KNX scene*** for the third).

6.21 *"6 videoentryphone events"* function

(For combined KNX/videoentryphone panels only)

This function is used to create up to 6 logical connections between the KNX system and the video entryphone system so there is interaction between the two. Therefore it will be possible to control video entryphone events via KNX BUS commands or to generate feedback on the KNX BUS following video entryphone events. End user side, the interactions between the two systems are "transparent".

The basic structure of the menu is as follows:

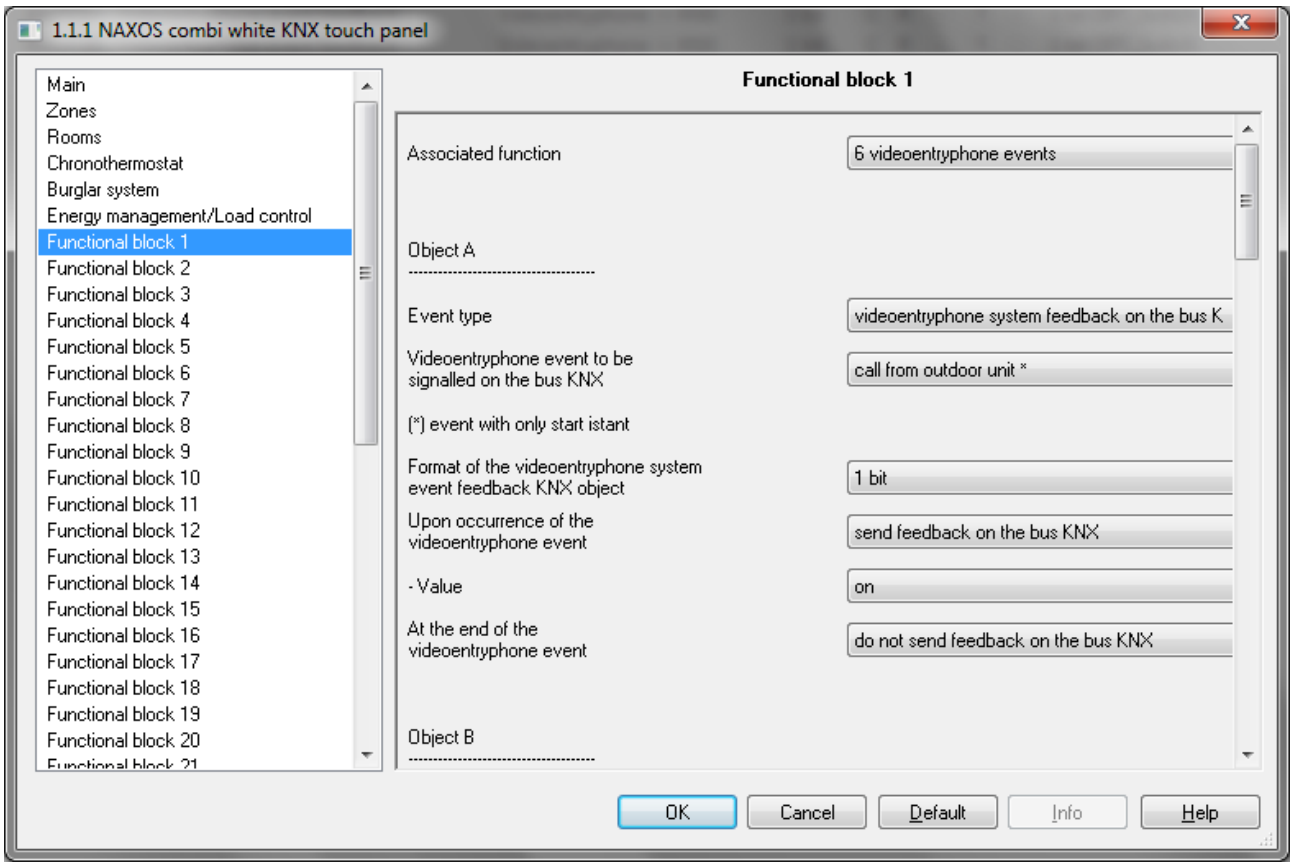


Fig 25: Setting ETS parameters - "6 videoentryphone events function" section

The parameter **"Event type"** is used to define the type of interaction to be implemented for each of the 6 available objects (A, B, C, D, E or F depending on the header to which the parameter refers), preparing the communication object to carry out this function and displaying the parameters necessary for defining this interaction in detail. the possible values are:

- **videoentryphone system feedback on the bus KNX** **(default value)**
- command from the bus KNX to the videoentryphone system

selecting **videoentryphone system feedback on the bus KNX**, displays the parameters **"Videoentryphone event to be signalled on the bus KNX"**, **"Format of the videoentryphone system event feedback KNX object"**, **"Upon occurrence of the videoentryphone event"** and **"At the end of the videoentryphone event"**.

Selecting **command from the bus KNX to the videoentryphone system** displays the parameters **"KNX command function to the video entryphone system"**, **"Associate the KNX command to"** and the communication object ***Block xA - Command to videoentyphone system*** (Date Point Type: 1.001 DPT_Switch), ***Block xB - Command to videoentyphone system*** (Date Point Type: 1.001 DPT_Switch), ***Block xC - Command to videoentyphone system*** (Date Point Type: 1.001 DPT_Switch), ***Block xD - Command to videoentyphone system*** (Date Point Type: 1.001 DPT_Switch), ***Block xE - Command to videoentyphone system*** (Date Point Type: 1.001 DPT_Switch) or ***Block xF - Command to***

videoentyphone system (Date Point Type: 1.001 DPT_Switch), depending on the parameter to which the object refers.

The parameter **"Videoentryphone event to be signalled on the bus KNX"**, which is displayed if the event type is "videoentryphone system feedback on the bus KNX", is used to define which video entryphone event will generate the sending of the KNX telegram via the associated communication object (A, B, C, D, E or F depending on the header to which the parameter refers). the possible values are:

- **call from outdoor unit * (default value)**
  call from outdoor unit in progress; it is not possible to determine which plate originated the call
- call from porter *
  call from porter in progress; it is not possible to determine which porter originated the call
- incoming intercom call
  call from intercom indoor unit in progress; it is not possible to determine which indoor position originated the call
- indication of missed call from outdoor unit *
  indication of missed call from outdoor unit
- indication of missed intercom call *
  indication of missed call from intercom indoor unit
- camera view command *
  the panel sent the camera view command to the video entryphone system; it is not possible to determine the outdoor unit of auto-insertion
- door opener command *
  the panel sent the door opener command to the video entryphone system
- auxiliary command 1 *
  the panel sent the auxiliary command 1 to the video entryphone system
- auxiliary command 2 *
  the panel sent the auxiliary command 2 to the video entryphone system
- intercom 1 outgoing call
  the panel sent the request for an intercom call to indoor unit 1
- intercom 2 outgoing call
  the panel sent the request for an intercom call to indoor unit 2
  ….

- intercom 10 outgoing call
  the panel sent the request for an intercom call to indoor unit 10
- response to a call *
  the panel responded to a call
- closing communication (call from outdoor unit) *
  the communication with an outdoor unit has been terminated

NB: These events are always available as "actions" of a sequence scene or timer variable without having to create an interaction between the KNX and video entryphone systems.
(*) these events do not have the end instant; any KNX feedback associated with end instant will never be sent.

The parameter **"Format of the videoentryphone system event feedback KNX object"** which is displayed if the event type is "videoentryphone system feedback on the bus KNX", is used to define the format and coding of the BUS telegram that will be sent by the panel via the associated communication object ***Block xA - Videoentyphone event feedback*** (A, B, C, D, E or F, depending on the header to which the parameter refers) after the occurrence of the video entryphone event. the possible values are:

- **1 bit (default value)**
  with this setting, the format of the above cited object is *1.001 DPT_Switch*
- 1 byte unsigned
  with this setting, the format of the above cited object is *5.010 DPT_Value_1_Ucount*
- 1 byte signed
  with this setting, the format of the above cited object is *6.010 DPT_Value_1_Count*
- 1 byte percentage
  with this setting, the format of the above cited object is *5.001 DPT_Scaling*
- 1 byte two's complement percentage

with this setting, the format of the above cited object is *6.001 DPT_Percent_V8*
- 1 byte HVAC mode
  with this setting, the format of the above cited object is *20.102 DPT_HVACMode*
- 1 byte scene
  with this setting, the format of the above cited object is *18.001 DPT_SceneControl*

Based on the value set for this item, both the format of the communication object dedicated to sending the feedback as well as the value range that can be sent upon occurrence of and at the end of the video entryphone event, will change.

The parameter **"Upon occurrence of the videoentryphone event"** is used to define if, upon occurrence of the video entryphone event, the device should or should not send a BUS telegram via the associated communication object (A, B, C, D, E or F depending on the header to which the parameter refers). In the same way, the parameter **"At the end of the videoentryphone event"** is used to define if, at the end of the video entryphone event, the device should or should not send a BUS telegram via the associated communication object (A, B, C, D, E or F depending on the header to which the parameter refers).

- If the format of the video entryphone event feedback object is **1 bit,** the values that can be set for the two parameters listed above are:

  - **do not send feedback on the bus KNX** **(default value at the end of the event)**
  - **send feedback on the bus KNX** **(default value upon occurrence of the event)**

  setting **send feedback on the bus KNX**, it is possible to define the value to be sent via the new displayed parameter **"Value"** which can have the following values:

  - **on** **(default value)**
  - off

- If the format of the video entryphone event feedback object is **1 byte unsigned**, the values that can be set for the two parameters listed above are:

  - **do not send feedback on the bus KNX** **(default value at the end of the event)**
  - **send feedback on the bus KNX** **(default value upon occurrence of the event)**

  By setting **send feedback on the bus KNX**, it is possible to define the value to be sent via the new displayed parameter **"Value (0 .. 255)"** which can assume the following values:

  - from **0 (default value)** to 255, with steps of 1

- If the format of the video entryphone event feedback object is **1 byte signed**, the values that can be set for the two parameters listed above are:

  - **do not send feedback on the bus KNX** **(default value at the end of the event)**
  - **send feedback on the bus KNX** **(default value upon occurrence of the event)**

  By setting **send feedback on the bus KNX**, it is possible to define the value to be sent via the new displayed parameter **"Value (-128 .. +127)"** which can assume the following values:

  - from -128 to 127 with steps of 1, **0 (default value)**

- If the format of the video entryphone event feedback object is **1 byte percentage**, the values that can be set for the two parameters listed above are:

  - **do not send feedback on the bus KNX** **(default value at the end of the event)**
  - **send feedback on the bus KNX** **(default value upon occurrence of the event)**

  By setting **send feedback on the bus KNX**, it is possible to define the value to be sent via the new displayed parameter **"Value (0% .. 100%)"** which can assume the following values:

- from **0% (default value)** to 100% with step of 1

- If the format of the video entryphone event feedback object is **1 byte percentage completion by two**, the values that can be set for the two parameters listed above are:

  - **do not send feedback on the bus KNX** **(default value at the end of the event)**
  - **send feedback on the bus KNX** **(default value upon occurrence of the event)**

  by setting **send feedback on the bus KNX**, it is possible to define the value to be sent via the new displayed parameter **"Value (-128% .. +127%)"** which can assume the following values:

  - from -128 to 127 with steps of 1, **0 (default value)**

- If the format of the video entryphone event feedback object is **1 byte HVAC mode**, the values that can be set for the two parameters listed above are:

  - **do not send feedback on the bus KNX** **(default value at the end of the event)**
  - **send feedback on the bus KNX** **(default value upon occurrence of the event)**

  setting **send feedback on the bus KNX**, it is possible to define the value to be sent via the new displayed parameter **"Value"** which can have the following values:

  - **auto** **(default value)**
  - comfort
  - precomfort
  - economy
  - off (building protection)

- If the format of the video entryphone event feedback object is **1 byte scene**, the values that can be set for the two parameters listed above are:

  - **do not send feedback on the bus KNX** **(default value at the end of the event)**
  - **send feedback on the bus KNX** **(default value upon occurrence of the event)**

  setting **send feedback on the bus KNX**, it is possible to define the value to be sent via the new displayed parameter **"Value"** which can have the following values:

  - **execute scene 0** **(default value)**
  - execute scene 1
  - ..
  - execute scene 63
  - learn scene 0
  - learn scene 1 9
  - ..
  - learn scene 63

The parameter **"Videoentryphone event to be controlled"**, which is displayed if the event type is "command from the bus KNX to the videoentryphone system", is used to define which video entryphone event will be controlled by the KNX BUS via the associated communication object (A, B, C, D, E or F depending on the header to which the parameter refers). The possible values are:

- camera view
  camera view can be activated from the KNX BUS; it is not possible to select a definite outdoor unit for auto-insertion
- door opener command *
  it is possible to send the door opener command from the KNX BUS; the door opener relay activated is the one for the outdoor unit in communication or the default one when the system is idle, as the recipient outdoor unit cannot be selected
- auxiliary 1 command *

it is possible to send the auxiliary command 1 from the KNX BUS
- auxiliary 2 command *
  it is possible to send the auxiliary command 2 from the KNX BUS
- videoentryphone voice mail
  it is possible to enable/disable the videoentryphone voice mail from the KNX BUS
- privacy (ringer off)
  it is possible to exclude/activate the videoentryphone ringer from the KNX BUS
- door call *
  it is possible to activate the ringer associated with the call from door call from the KNX BUS; function equivalent to closing the clean "bell" contact on the panel terminal block
- office function
  from the KNX bus, you can activate/deactivate the office function (function that allows the opening of the door and the camera view automatically without videoentrphone ringtone)

(*) these events do not have the deactivation instant; any KNX command associated with deactivation instant will not have any effect.

The parameter **"Associate the event to the value"** is used to define the type of BUS command received via the dedicated object (A, B, C, D, E or F depending on the header to which the parameter refers) which triggers the start and end of the associated video entryphone event. The values that can be set are:

- "0" -> activation
- **"1"** -> **activation** **(default value)**
- "1" and "0" -> activation
- "1" -> activation / "0" -> deactivation
- "0" -> activation / "1" -> deactivation"

7 "*Chronothermostat*" menu

The **Chronothermostat** menu contains only those parameters that allow you to enable and configure the operating parameters for each of the 4 available chronothermostat functions. The panel implements the chronothermostat function if the measurement of the temperature of the zone to control is made available via an external sensor on the KNX BUS.

The basic structure of the menu is as follows:

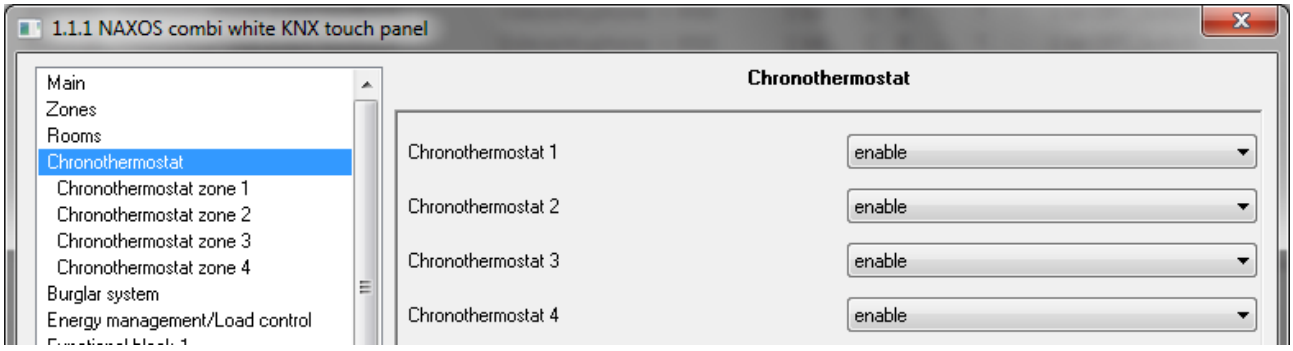


Fig 26: Setting ETS parameters - "Chronothermostat" section

The parameters **"Chronothermostat 1"**, **"Chronothermostat 2"**, **"Chronothermostat 3"** and **"Chronothermostat 4"** are used to display and configure all the operating parameters for the relative chronothermostat functions grouped in the menus **Chronothermostat zone 1**, **Chronothermostat zone 2**, **Chronothermostat zone 3** and **Chronothermostat zone 4**. The values that can be set for these parameters are:

- **disable** **(default value)**
- enable

when the **enable** value is set, the corresponding configuration menus are visible.

If “Chronothermostat 1” is enabled, the menus **Functional block 16** and **Functional block 17** are no longer displayed.
If “Chronothermostat 2” is enabled, the menus **Functional block 20** and **Functional block 21** are no longer displayed.
If “Chronothermostat 3” is enabled, the menus **Functional block 24** and **Functional block 25** are no longer displayed.
If “Chronothermostat 4” is enabled, the menus **Functional block 28** and **Functional block 29** are no longer displayed.

7.1 “Chronothermostat zone x” menu

For the sake of simplicity, the items that make up the menus **Chronothermostat zone 1**, **Chronothermostat zone 2**, **Chronothermostat zone 3** and **Chronothermostat zone 4** will be described only once in the following chapters (in reference to the generic menu **Chronothermostat zone x**) as all these menus have the same items.
The **Chronothermostat zone x** menu presents the parameters used to customise both the graphic element that identifies the chronothermostat object as well as the operating rules for the object itself.

The graphic element that represents this function, inside the navigation tree belongs to the “Temperature adjustment” category.

The basic structure of the menu is as follows:

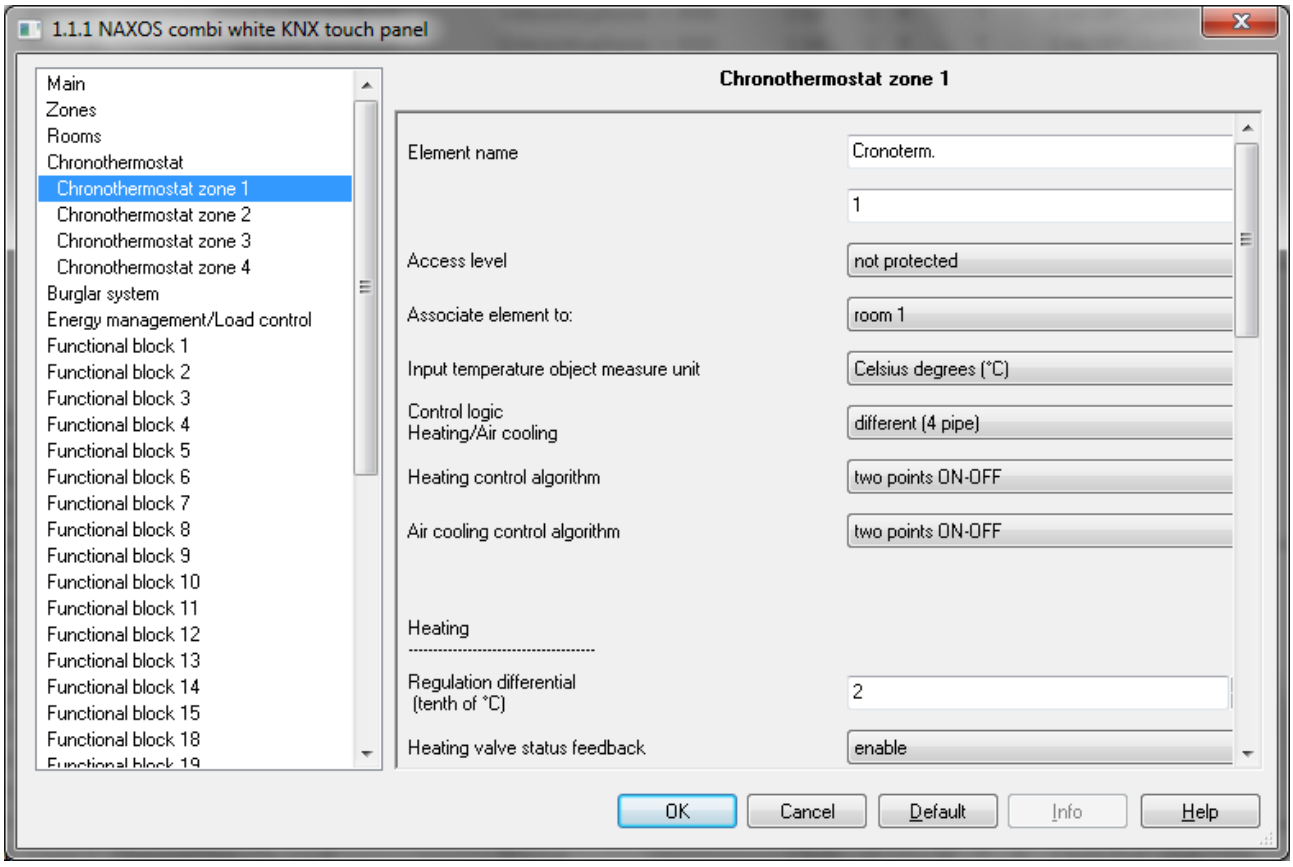


Fig 27: Setting ETS parameters - “Chronothermostat zone X” section

The parameter **"Element name"** is used to enter the name you want to associate with the chronothermostat zone x element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for accessing the commands available for the element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the navigation structure, each element of this type must be associated with an Room; this association is defined via the parameter **"Associate element to"**, which can have the following values:

- **room 1** **(default value)**
- room 2
- room 3
- ……
- room 31
- room 32

The parameter **"Input temperature measure unit"** is used to set the measure unit with which the information received via the communication object ***Chrono x - Measured temperature input*** will be decoded; this information is used by the chronothermostat function as the room temperature to be controlled, as the panel does not have room temperature sensors. The values that can be set are:

- **Celsius degrees (°C)** **(default value)**
- Kelvin degrees (°K)
- Fahrenheit degrees (°F)

the value set for this parameter changes the coding of the communication object ***Chrono x - Measured temperature input***: *9.001 DPT_Value_Temp* if the value is **Celsius degrees (°C)**, *9.002 DPT_Value_Tempd* if the value is **Kelvin degrees (°K)** and *9.027 DPT_Value_Temp_F* if the value is **Fahrenheit degrees (°F)**.

Given the different types of temperature adjustment systems, it is possible to dedicate a common solenoid valve control object to the heating and air cooling system or dedicate one to each of the two types of operation. The parameter **"Control logic Heating/Air cooling"** is used to define if the chronothermostat function x control logic, and as a result the control communication object, is common for the heating and air cooling or if it is different; the values that can be set are:

- common (2 pipe)
- **different (4 pipe) (default value)**

select **common (2 pipe)** to view the parameters **"Heating/air cooling control algorithm"** and **"Heating/air cooling valve status feedback"**, whereas select **different (4 pipe)** to view the parameters **"Heating control algorithm"** and **"Air cooling control algorithm"**.

The parameter **"Heating control algorithm"** is used to define the control algorithm used for the heating system; the values that can be set are:

- **two points ON-OFF** **(default value)**
- proportional-integral PWM
- fan coil with ON-OFF speed control

select the value **two points ON-OFF** to display the parameters **"Regulation differential (tenth of °C)"** and **"Heating valve status feedback"** in the section **Heating** and the communication object ***Chrono x - Heating valve switch*** (Date Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **proportional-integral PWM** to view the parameters **"Select heating system"**, **"Proportional band"**, **"Integration time" "Cycle time"** and **Heating valve status feedback** in the **Heating** section and the communication object ***Chrono x - Heating valve switch*** (Date Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **fancoil with ON-OFF speed control** to view the parameters **"Speed 1 regulation differential (tenth of °C)"**, **"Speed 2 regulation differential (tenth of °C)"**, **"Speed 3 regulation differential (tenth of °C)"**, **"Speed 1 inertia time (seconds)"**, **"Speed 2 inertia time (seconds)"**, **"Speed 3 inertia time (seconds)"**, **"Fancoil speed status feedback"**, **"Valve regulation differential (tenth of °C)"** and **"Heating valve status feedback"** in the **Heating** section and the communication objects ***Chrono x - V1 fan switching heating***, ***Chrono x - V2 fan switching heating***, ***Chrono x - V3 fan switching heating*** and ***Chrono x - Heating valve switch*** (Date Point Type: 1.001 DPT_Switch) for respectively controlling the first, second and third speed of the fancoil and the heating solenoid valve.

The parameter **"Air cooling control algorithm"** is used to define the control algorithm used for the air cooling system; the values that can be set are:

- **two points ON-OFF** **(default value)**
- proportional-integral PWM
- fan coil with ON-OFF speed control

select the value **two points ON-OFF** to display the parameters **"Regulation differential (tenth of °C)"** and **"Air cooling valve status feedback"** in the section **Air cooling** and the communication object ***Chrono x - Cooling valve switch*** (Date Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **proportional-integral PWM** to view the parameters **"Select air cooling system"**, **"Proportional band"**, **"Integration time" "Cycle time"** and **Air cooling valve status feedback** in the **Air cooling** section and the communication object ***Chrono x - Cooling valve switch***(Date Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **fancoil with ON-OFF speed control** to view the parameters **"Speed 1 regulation differential (tenth of °C)"**, **"Speed 2 regulation differential (tenth of °C)"**, **"Speed 3 regulation differential (tenth of °C)"**, **"Speed 1 inertia time (seconds)"**, **"Speed 2 inertia time (seconds)"**, **"Speed 3 inertia time (seconds)"**, **"Fancoil speed status feedback"**, **"Valve regulation differential (tenth of °C)"** and **"Air cooling valve status feedback"** in the **Air cooling** section and the communication objects ***Chrono x - V1 fan switching cooling***, ***Chrono x - V2 fan switching cooling***, ***Chrono x - V3 fan switching cooling*** and ***Chrono x - Cooling valve switch*** (Date Point Type: 1.001 DPT_Switch) for respectively controlling the first, second and third speed of the fancoil and the air cooling solenoid valve.

The parameter **"Heating/air cooling control algorithm"** is used to define the control algorithm used both for the heating system as well as for the air cooling system, as the control logic is common; the values that can be set are:

- **two points ON-OFF** **(default value)**
- proportional-integral PWM
- fan coil with ON-OFF speed control

select the value **two points ON-OFF**, in the sections **Heating** and **Air cooling** to display the parameters **"Regulation differential (tenth of °C)"** and the communication object ***Chrono x - Heating/Cooling valve switch*** (Date Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select the value **proportional-integral PWM**, in the sections **Heating** and **Air cooling** to display the parameters **"Select heating system** (**Air cooling** in the **Air cooling)"** section, **"Proportional band"**, **"Integration time"** and **"Cycle time"** and the communication object ***Chrono x - Heating /Cooling valve switch*** (Date Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **fancoil with ON-OFF speed control** to display in the sections **Heating** and **Air cooling** the parameters **"Speed 1 regulation differential (tenth of °C)"**, **"Speed 2 regulation differential (tenth of °C)"**, **"Speed 3 regulation differential (tenth of °C)"**, **"Speed 1 inertia time (seconds)"**, **"Speed 2 inertia time (seconds)"**, **"Speed 3 inertia time (seconds)"**, **"Fancoil speed status feedback** and **"Valve regulation differential (tenth of °C)"** and the communication objects ***Chrono x - V1 fan switching heating***, ***Chrono x - V2 fan switching heating***, ***Chrono x - V3 fan switching heating***, ***Chrono x - V1 fan switching cooling***, ***Chrono x - V2 fan switching cooling*** and ***Chrono x - V3 fan switching cooling*** and

Chrono x - Heating/Cooling valve switch (Date Point Type: 1.001 DPT_Switch) for respectively controlling the first, second and third speed of fancoil for air cooling and the common solenoid valve.

The parameter **"Valve regulation differential (tenth of °C)"** is used to set the regulation differential value of the 2 points control of the fancoil operating solenoid valve for heating and air cooling, depending on the section to which the parameter refers. When heating, this value, which was already mentioned in the *Control algorithms* paragraph, when subtracted from the value of the setpoint, determines the value of the threshold below which the heating system valve is activated in the fancoil mode. When cooling, adding this value to the setpoint value determines the value of the threshold above which the air cooling system valve is activated. The values that can be set are:

- from 1 to 20 with step of 1 **(default value)**

The parameter **"Heating/air cooling valve status feedback"** is used to enable the device to receive feedback from the actuator that commands the heating/air cooling solenoid valve; in this way, the device is able to receive the telegram after the solenoid valve switched and to repeat the command if the switching did not take place. The values that can be set are:

- disable
- **enable (default value)**

select **disable** to view the parameter **"Command repetition period with disabled feedback"**; selecting the value **enable** displays the communication object ***Chrono x - Heating/cooling valve status feedback*** (Date Point Type: 1.001 DPT_Switch). When BUS voltage is restored, the device sends the read request command via the object ***Chrono x - Heating/cooling valve status feedback*** to be updated about the status of the heating/air cooling solenoid valve.

During normal operation of the temperature adjustment, the actuator status can be changed by an entity external of the panel, that forces its status, modifying it. In this case, the device repeats the valve switching command to realign the status of the actuator with the one determined by the control logic of the chronothermostat.

With the solenoid valve status feedback disabled, it may be useful to cyclically repeat the command to the actuator that manages the solenoid valve so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The parameter **"Command repetition period with disabled feedback"** is used to define the frequency of the cyclical sending; the values that can be set are:

- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes (default value)**

The parameter **"Regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the **two points ON-OFF** control algorithm for heating and air cooling, depending on the section to which the parameter refers. When heating, this value, which was already mentioned in the *Control algorithms* paragraph, when subtracted from the value of the setpoint, determines the value of the threshold below which the heating system is activated in the 2-point control mode. When cooling, adding this value to the setpoint value determines the value of the threshold above which the air cooling system is activated. The values that can be set are:

- from 1 to 20 with step of 1, **2 (default value)**

The parameter **"Select heating system"** is used to automatically measure the operating parameters (Proportional band and Integration time) of the proportional integral algorithm based on the selected heating system. The values that can be set are:

- hot water heating
- **floor heating (default value)**
- fancoil

- electrical heating
- personalized

selecting **hot water heating**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified and the values **3.5 °C** and **150 minutes** are displayed.
Selecting **floor heating**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified, and the values **3.5 °C** and **240 minutes** are displayed.
Selecting **fancoil**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified, and the values **3.0 °C** and **90 minutes** are displayed.
Selecting **electrical heating**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified, and the values **3.0 °C** and **100 minutes** are displayed.
Selecting **personalized**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified.

The parameter **"Select air cooling system"** is used to automatically measure the operating parameters (Proportional band and Integration time) of the proportional integral algorithm based on the selected air cooling system. The values that can be set are:

- **ceiling cooling (default value)**
- fancoil
- personalized

Selecting **ceiling cooling**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified, and the values **5.0 °C** and **240 minutes** are displayed.
Selecting **fan coil unit**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified, and the values **4.0 °C** and **90 minutes** are displayed.
Selecting **personalized**, the parameters **"Proportional band"** and **"Integration time"** will be displayed but cannot be modified.

The parameter **"Proportional band"** is used to set the width of the PWM proportional band of the **proportional-integral PWM** control algorithm for heating or air cooling, depending on the section to which the parameter refers. When heating, this value, which was already mentioned in the *Control algorithms*, when subtracted from the value of the setpoint, determines the lower limit of the proportional band used for the proportional integral control. When cooling, this value, when added to the setpoint value determines the upper limit of the proportional band used for the proportional integral control. The values that can be set are:

- 1.0 °C
- 1.1 °C
- …
- **2.0 °C (default value)**
- …
- 10.0°C

The parameter **"Integration time"** is used to set the contribution of the integral action in the proportional integral control (see the *Control algorithms* paragraph) for heating or air cooling, depending on the section to which the parameter refers. The values that can be set are:

- from 1 minute to 250 minutes with step of 1 plus the value "no integral", **60 (default value)**

Selecting **no integral**, the integral component is zero and the pure effect of proportional control is obtained.

The parameter **" Cycle time"** is used to set the value of the period within which the device carries out PWM modulation, modifying the duty-cycle of heating or air cooling, depending on the section to which the parameter refers. The values that can be set are:

- from 5 minutes to 60 minutes with step of 1 minute, **30 minutes (default value)**

The parameter **"Speed 1 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the first speed of the **fancoil with ON-OFF speed control** algorithm for heating and air cooling, depending on the section to which the parameter refers. When heating, this value, which was

already mentioned in the *Control algorithms* paragraph, when subtracted from the value of the "setpoint-$_{\Delta}T_{valv}$" determines the threshold value under which fancoil speed 1 is activated. When cooling, this value, when added to the value "setpoint+$_{\Delta}T_{valv}$", determines the threshold value under which fancoil speed 1 is activated. The values that can be set are:

- from 0 to 20 with step of 1 **(default value)**

setting the value **0** when heating obtains the condition "$_{\Delta}T_{1\ heat=\Delta}T_{valv}$" therefore the value of the speed 1 activation threshold is "setpoint-$_{\Delta}T_{valv}$" and the deactivation value is "setpoint"; when cooling, the condition "$_{\Delta}T_{1\ cond=\Delta}T_{valv}$" is obtained, therefore the value of the speed 1 activation threshold is "setpoint+$_{\Delta}T_{valv}$" and the deactivation value is "setpoint".

The parameter **"Speed 2 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the second speed of the **fancoil with ON-OFF speed control** algorithm for heating and air cooling, depending on the section to which the parameter refers. When heating, this value, which was already mentioned in the *Control algorithms* paragraph, when subtracted from the value "setpoint-$_{\Delta}T_{valv\Delta}T_{1\ heat}$ " determines the threshold value under which fancoil speed 2 is activated. When cooling, this value, when subtracted from the value "setpoint+$_{\Delta}T_{valv+\Delta}T_{1\ cond}$", determines the threshold value under which fancoil speed 2 is activated. The values that can be set are:

- from 1 to 20 with step of 1 **(default value)**

The parameter **"Speed 3 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the third speed of the **fancoil with ON-OFF speed control** algorithm for heating and air cooling, depending on the section to which the parameter refers. When heating, this value, which was already mentioned in the *Control algorithms* paragraph, when subtracted from the value "setpoint-$_{\Delta}T_{valv-\Delta}T_{1\ heat-\Delta}T_{2\ heat}$" determines the threshold value under which fancoil speed 3 is activated. When cooling, this value, when subtracted from the value "setpoint+$_{\Delta}T_{valv+\Delta}T_{1\ cond+\Delta}T_{2\ cond}$", determines the threshold value under which fancoil speed 3 is activated. The values that can be set are:

- from 1 to 20 with step of 1 **(default value)**

When, according to the "fancoil with speed control" algorithm, the device must turn on any speed and speed 1 is on, a delay can be inserted between the moment in which feedback is received that speed 1 is turned off (or the moment the command to turn off speed 1 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **"Speed 1 inertia time (seconds)"** is used to define the extent of the delay between the deactivation of speed 1 and the activation of the new speed for heating or air cooling, depending on the section to which the parameter refers. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

When, according to the "fancoil with speed control" algorithm, the device must turn on any speed and speed 2 is on, a delay can be inserted between the moment in which feedback is received that speed 2 is turned off (or the moment the command to turn off speed 2 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **"Speed 2 inertia time (seconds)"** is used to define the extent of the delay between the deactivation of speed 1 and the activation of the new speed for heating or air cooling, depending on the section to which the parameter refers. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

When, according to the "fancoil with speed control" algorithm, the device must turn on any speed and speed 3 is on, a delay can be inserted between the moment in which feedback is received that speed 3 is turned off (or the moment the command to turn off speed 3 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **"Speed 3 inertia time (seconds)"** is used to define the extent of the delay between the deactivation of speed 3 and the activation of the new speed for heating or air cooling, depending on the section to which the parameter refers. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

Defining the inertia times is useful for preserving the integrity of the fancoil, because the fact of turning off the power supply to the motor (turning off the actuator) of a fancoil speed does not guarantee that current is no longer circulating in the winding and the instantaneous supply of power to another winding could damage the fancoil (simultaneous powering of multiple windings).

The parameter **Heating valve status feedback** or **"Air cooling valve status feedback"** is used to enable the device to receive feedback from the actuator that commands the heating or air cooling solenoid valve; in this way, the device is able to receive the telegram after the solenoid valve switched and to repeat the command if the switching did not take place. The values that can be set are:

- disable
- **enable (default value)**

select **disable** to view the parameter **"Command repetition period with disabled feedback"**; selecting the value **enable** displays the communication object ***Chrono x - Heating valve status feedback*** (Date Point Type: 1.001 DPT_Switch) for heating or ***Chrono x - Cooling valve status feedback*** (Date Point Type: 1.001 DPT_Switch) for air cooling. When BUS voltage is restored, the device sends the read request command via the object ***Chrono x - Heating valve status feedback*** or ***Chrono x - Cooling valve status feedback*** to be updated about the status of the heating solenoid valve.

During normal operation of the temperature adjustment, the actuator status can be changed by an entity external of the panel, that forces its status, modifying it. In this case, the device repeats the valve switching command to realign the status of the actuator with the one determined by the control logic of the chronothermostat. In the same manner, if the control algorithm is operating in heating mode and feedback is received that the air cooling valve is activated, the algorithm is suspended immediately while the command for deactivating the air cooling solenoid valve is sent (triggering the process for waiting for confirmation and repeating the command until the confirmation is received) until the problem is resolved.

With the heating or air cooling solenoid valve status feedback disabled, it may be useful to cyclically repeat the command to the actuator that manages the solenoid valve so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The parameter **"Command repetition period with disabled feedback"** is used to define the time range of the cyclical sending; the values that can be set are:

- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes (default value)**

If the control algorithm is fancoil, more important than the valve feedback is the possibility to receive feedback about the fancoil speed ON status. By enabling feedback, the device is always aware of the status of the speeds it commands. As the system does not always have actuators dedicated to the fancoil with mechanically interlocked outputs, the logical interlock function must be implemented on a firmware level which makes it possible to turn on a fancoil speed that is different than what is on only if the correct feedback is received from the latter that it was turned off (providing speed feedback is enabled); as long as the panel does not receive feedback that the active speed was turned off, it will not send the command to turn on the new speed to prevent multiple fancoil windings from being supplied with power at the same time, which would break the fancoil. The parameter **"Fancoil speed status feedback"** is used to enable the device to receive feedback from the actuator that commands the fancoil speeds. The values that can be set are:

- disable
- **enable (default value)**

Select **disable** to view the parameter **"Fancoil speed command repetition period"**; selecting **enable** displays the communication objects ***Chrono x - Heating fan V1 status feedback***, ***Chrono x - Heating fan V2 status feedback*** *e* ***Chrono x - Heating fan V3 status feedback*** (Date Point Type: 1.001 DPT_Switch) when heating or ***Chrono x - Cooling fan V1 status feedback***, ***Chrono x - Cooling fan V2 status feedback*** and ***Chrono x - Cooling fan V3 status feedback*** (Date Point Type: 1.001 DPT_Switch) when cooling.

When BUS voltage is restored, the device sends the read request command via the objects ***Chrono x - Heating fan V1 status feedback***, ***Chrono x - Heating fan V2 status feedback*** and ***Chrono x - Heating fan V3 status feedback*** when heating or ***Chrono x - Cooling fan V1 status feedback***, ***Chrono x - Cooling fan V2 status feedback*** and ***Chrono x - Cooling fan V3 status feedback*** when cooling to be updated about the fancoil speed activation status.
If the fancoil feedback is disabled, deactivation commands of the inactive speeds must be sent for every speed activation command; in the same manner, every speed deactivation command must be sent together with deactivation commands for the other speeds.

The parameter **"Fancoil speed command repetition period"** is used to define the time range of the cycling sending to the fancoil speeds; the values that can be set are:

- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes (default value)**

The commands are repeated on all speed communication objects.

The parameter **"HVAC mode feedback"** is used to set the conditions for sending the HVAC mode feedback via the communication object ***Chrono x - HVAC mode feedback*** with a 1 byte size. The values that can be set are:

- **disabled (default value)**
- send on demand only
- send in case of change

selecting **on demand only** value, the HVAC mode feedback will not be sent spontaneously by the device via the communication object ***Chrono x - HVAC mode feedback*** but in the case of a status reading request, it sends the requester a telegram in response to the received command, which includes information about the HVAC mode set on the device. Selecting **on variation** value, the HVAC mode feedback will be sent spontaneously by the device via the communication object ***Chrono x - HVAC mode feedback*** each time the mode is changed. After a BUS voltage recovery, feedback about the active mode should be sent in order to update any connected devices. When the profile is activated, the signalled mode is AUTO.

The parameter **"Functioning type feedback"** is used to enable and set the conditions for sending feedback about the functioning type (Heating/Air cooling) set for the device chronothermostat function via the BUS telegram on the communication object ***Chrono x - Functioning type feedback*** (Date Point Type: 1.100 DPT_Heat/Cool). The values that can be set are:

- **disabled (default value)**
- send on demand only
- send in case of change

selecting **on demand only**, the feedback about the type of operation set on the device will not be sent spontaneously by the device via the communication object ***Chrono x - Functioning type feedback*** but in the case of a status reading request, it sends the requester a telegram in response to the received command, which includes information about the type of operation set on the device. Selecting **send on variation**, the feedback about the type of operation set on the device is sent spontaneously by the device via the communication object ***Chrono x - Functioning type feedback*** each time the functioning type changes. After a BUS voltage recovery, feedback about the type of active operation should be sent in order to update any connected devices.
The parameter **Current setpoint feedback** is used to enable and set the conditions for sending the feedback regarding the current setpoint value via the BUS telegram on the communication object ***Chrono x - Current setpoint feedback*** (Date Point Type: 9.001 DPT_Temp if object in °C, 9.002 DPT_Tempd if object in °K and 9.*027 DPT_Value_Temp_F* if object in °F). The values that can be set are:

- **disabled** **(default value)**
- send object (°C) on demand only
- send object (°K) on demand only

- send object (°F) on demand only
- send object (°C) in case of change
- send object (°K) in case of change
- send object (°F) in case of change

selecting **send object in (°C) on demand only**, **send object in (°K) on demand only** or **send object in (°F) on demand only**, the feedback regarding the active setpoint on the device is not sent spontaneously by the device via the communication objects ***Chrono x - Current setpoint feedback*** but in the case of a status reading request, it sends the requester a telegram in response to the received command, which includes information about the setpoint set on the device. By selecting **send object in (°C)** in case of change, **send object in (°K)** in case of change or **send object in (°F)** in case of change, the feedback regarding the active setpoint on the device is sent spontaneously by the device via the communication object ***Chrono x - Current setpoint feedback***, each time there is a variation in the setpoint itself (also following a temporary forcing). After a BUS voltage recovery, feedback about the active setpoint should be sent in order to update any connected devices.

Management of the HVAC operating mode (Auto/Comfort/Precomfort/Economy/Off) and the temperature adjustment operating type (heating/air cooling) of the chronothermostat function is managed locally by the relative graphic element, but it can also be remotely modified; the HVAC mode can be modified via the communication object ***Chrono x - HVAC mode input*** (Date Point Type: 20.102 DPT_HVACMode) whereas the type of operation can be modified via the communication object ***Chrono x - Functioning type input*** (Date Point Type: 1.100 DPT_Heat/Cool).

If there is a KNX humidity sensor in the room controlled by the chronothermostat function, the graphic element that represents the chronothermostat displays the relative humidity of the room via the communication object ***Chrono x - Measured humidity input*** (Date Point Type: 9.007 DPT_Value_Humidity).

The scene feature allows you to issue two possible commands to the chronothermostat:

- execute scene, i.e. a command to move into a condition determined
- learning scene, i.e. a storage command of the current state (the instant in which the command is received) of the HVAC mode and the functioning type active

There are 4 different scenes, for which the chronothermostat can learn/execute four different conditions of the HVAC mode and functioning type.
Through the parameter **"Scene number 1"**, **"Scene number 2"**, **"Scene number 3"** and **"Scene number 4"** you can set the numeric value that is used to identify and consequently execute/learn the scene; the values that can be set are:

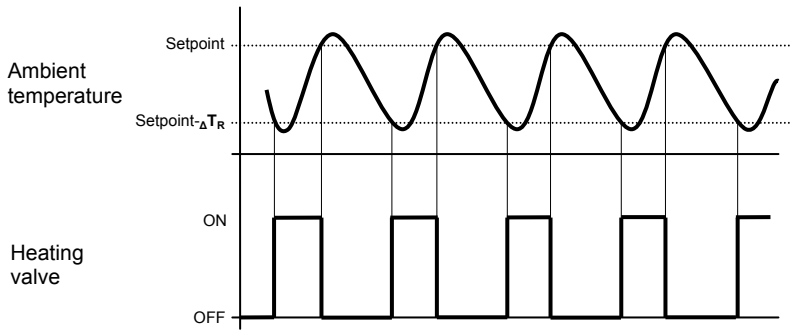
- **unassigned** **(default value)**
- 0, 1.. 63

7.2 Control algorithms

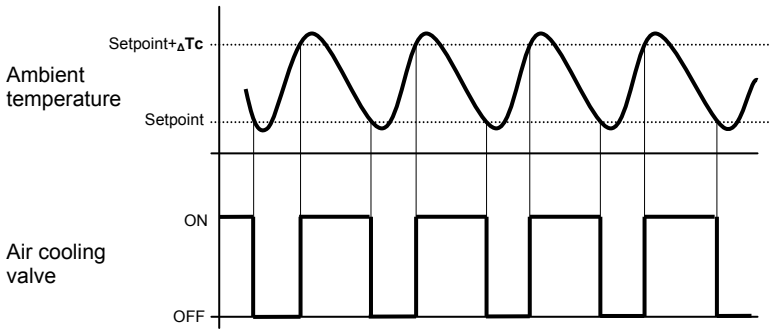
Independently of the fact if the control logic is common or different between the two operating types, the logic is as follows depending on the selected algorithm:

- ***two points ON-OFF***

The algorithm used for controlling the temperature adjustment system is the classic type that is called 2 points control. This type of control involves the turning on and off of the temperature adjustment system following a hysteresis cycle. This means that there is not a single threshold that discriminates between the turning on and off of the system, but there are two.



When the measured temperature is lower than the value setpoint-$_{\Delta}T_R$" (where $_{\Delta}T_R$ identifies the value of the heating regulation differential) the device turns on the heating system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches the fixed setpoint value, the device turns off the heating system, sending the relative BUS command to the BUS that manages it.
This makes it clear that there are two decision thresholds for turning the heating system on and off, the first consists of the value "setpoint- $_{\Delta}T_R$" below which the device turns on the system, the second consists of the setpoint value that was set, above which the device turns off the system.



When the measured temperature is higher than the value "setpoint+ $_{\Delta}$Tc" (where $_{\Delta}Tc$ identifies the air cooling regulation differential) the device turns on the air cooling system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches the fixed setpoint value, the device turns off the air cooling system, sending the relative BUS command to the BUS that manages it.
This makes it clear that there are two decision thresholds for turning the air cooling system on and off, the first is the setpoint value that was set, below which the device turns off the system, the second is the value "setpoint+$_{\Delta}$Tc" above which the device turns on the system.

- ***proportional-integral PWM***

The algorithm used to control the temperature adjustment system allows you to drastically reduce the times subject to thermal inertia and introduced by the 2 points control, called PWM control. This type of control involves the modulation of the impulse duty-cycle, represented by the temperature adjustment system activation time, on the basis of the difference between the fixed set-point and the temperature effectively

detected. Two components are needed to calculate the output function: the proportional component and the integral component.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

Proportional component

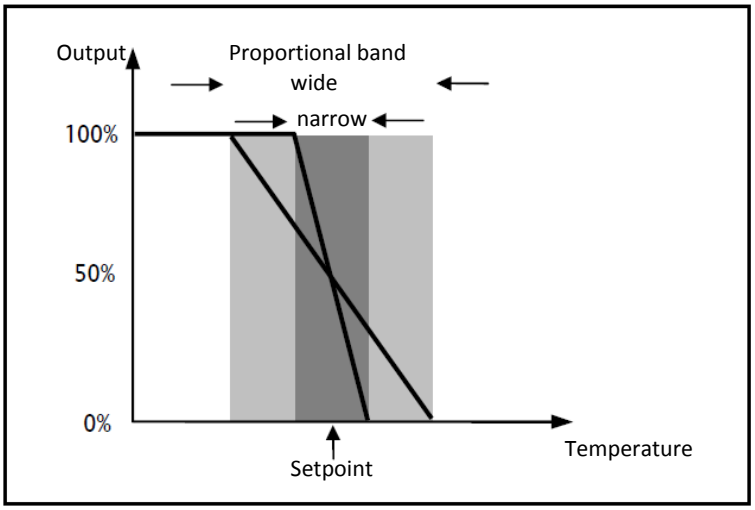
In the proportional component, the output function is proportional to the error (difference between setpoint and measured temperature).

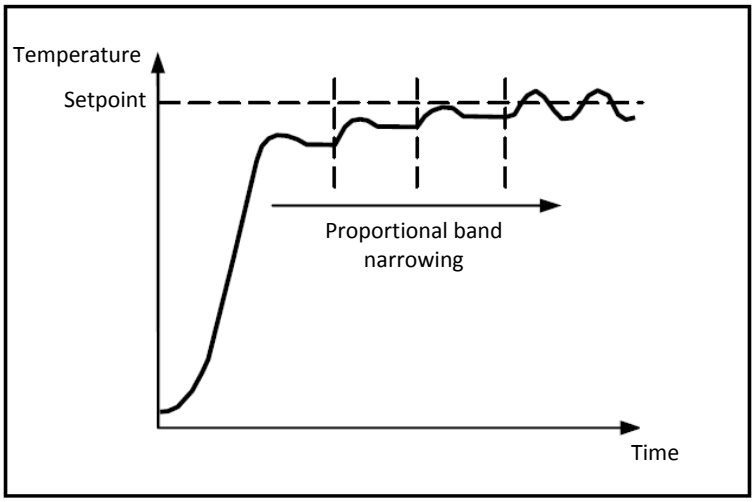
$$P_{out} = K_p e(t)$$

Once the proportional band is defined, the duty-cycle within the band varies between 0% and 100%; outside of the band, the duty-cycle will be maximum or minimum depending on the reference limits.

The width of the proportional band determines the extent of the response to the error. If the band is too “narrow”, the system will oscillate as it becomes more reactive; if the band is too "wide" the control system is slow. The ideal situation is when the proportional band is as narrow as possible without causing oscillations.

The diagram below shows the effect of narrowing the proportional band until the oscillation point of the output function. A "wide" proportional band results as a straight line in the control, but with an initial error between the setpoint and the actually perceptible temperature. As the band becomes narrower, the temperature approaches the reference value (setpoint) until it becomes unstable and starts to oscillate around it.

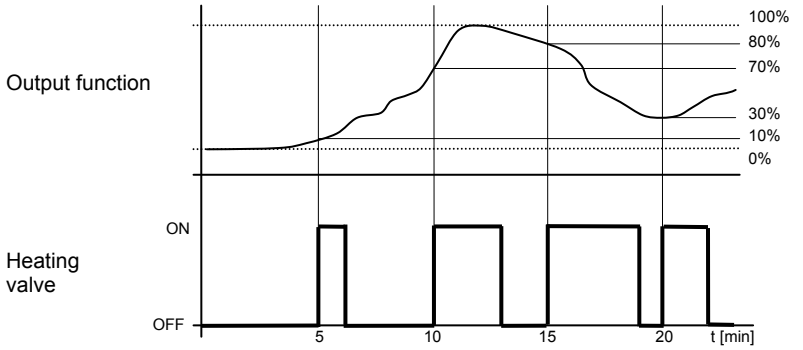




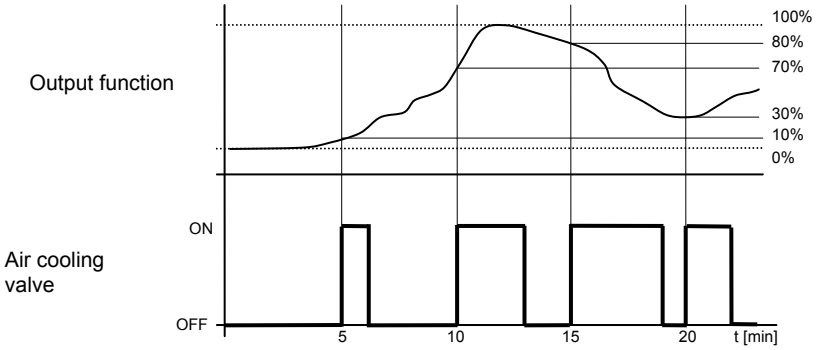
Integral component
The contribute of the integral period is proportional to the error (difference between the setpoint and the measured temperature) and its duration. The integral is the sum of the instantaneous error for every moment of time and provides the accumulated offset that should have been previously corrected. The accumulated error is then added to the regulator output.

$$I_{out} = K_i \int_0^t e(\tau)\, d\tau$$

The integral period accelerates the dynamics of the process towards the setpoint and eliminates the residuals of the stationary error status that takes place with a pure proportional controller.
The integration time is the parameter that determines the action of the integral component. The longer the integration time, the slower the modification of the output and hence the slower the system response. If the time is too short, the threshold value will be exceeded (overshoot), and the function will swing around the set-point.



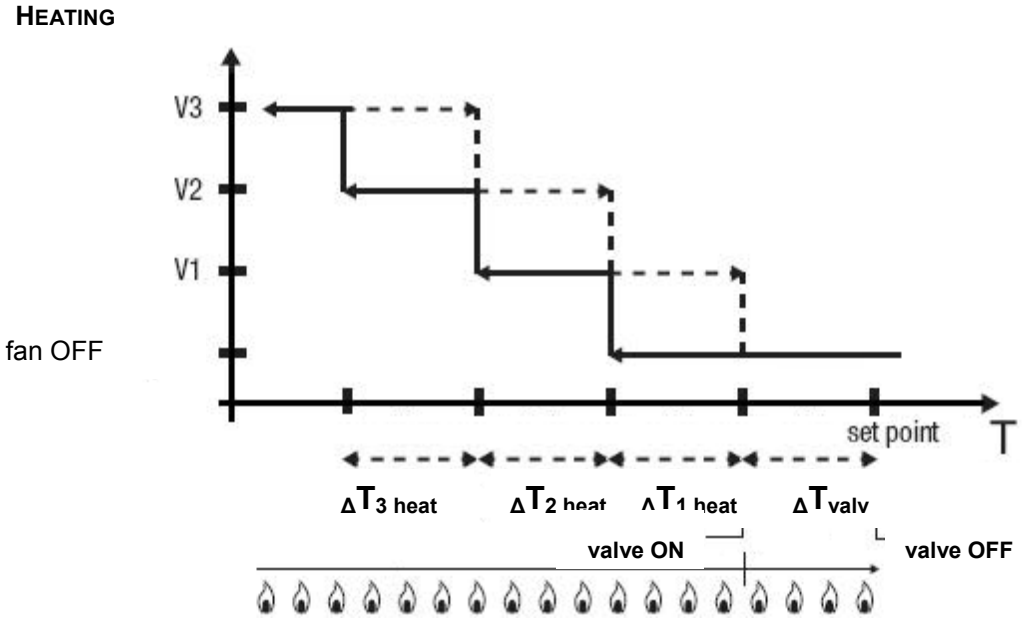
The device keeps the heating system switched on for a cycle time percentage that depends on the output function of the proportional-integral control; the device continuously regulates the heating system, modulating the system turning on-off times with a duty-cycle (shown to the right along the vertical axis) that depends on the output function value calculated at every time range equal to the cycle time. The cycle time is reinitialised every time the reference set-point is modified.
With this type of algorithm, there is no longer a hysteresis cycle on the heating device, so the inertia times (system heating and air cooling times) introduced by the 2 points control are eliminated. This produces energy savings because the system does not remain switched on when it is not needed and, once the required temperature has been reached, it continues to provide a heat limited contribution to compensate for the room heat dispersion.



As seen in the figure, the device keeps the air cooling system switched on for a cycle time percentage that depends on the output function of the proportional-integral control; the device continuously regulates the air cooling system, modulating the system turning on-off times with a duty-cycle (shown to the right along the vertical axis) that depends on the output function value calculated at every time range equal to the cycle time. The cycle time is reinitialised every time the reference set-point is modified.
With this type of algorithm, there is no longer a hysteresis cycle on the air cooling device, so the inertia times (system air cooling and heating times) introduced by the 2 points control are eliminated. This produces energy savings because the system does not remain switched on when it is not needed and, once the required temperature has been reached, it continues to provide a limited contribution of cold air to compensate for the contribution of room heat.

- ***fan coil with ON-OFF speed control***

The type of control that is applied when the fancoil control is enabled is similar to the 2 points control analysed in previous sections, which is to turn the fancoil speed on /off based on the difference between the setpoint that was set and the measured temperature.
The substantial difference with the 2-point algorithm is that, in this case, there is only one stage on which the hysteresis cycle is carried out, fixing the speed on and off thresholds, but there can be three; substantially, this means that each stage corresponds to a speed and when the difference between the measured temperature and the setpoint that was set causes a certain speed to be turned on, this means that before turning on the new speed, the other two must absolutely be turned off.

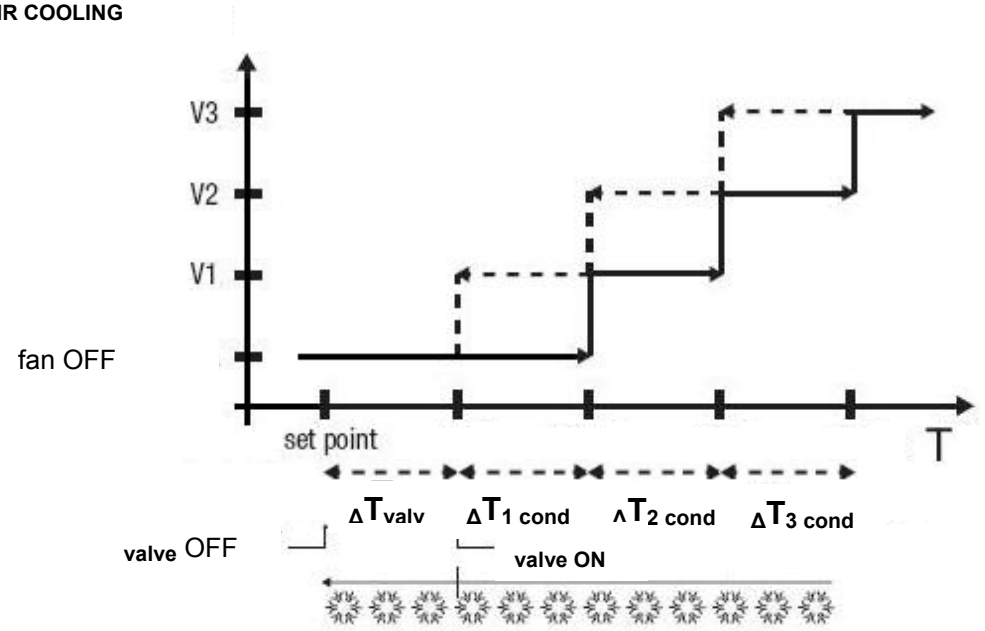


The figure refers to the control of the fancoil speeds with three operating stages for heating. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine its activation and deactivation. The thresholds are determined by values set for the various regulation differentials, and can be summarised as follows:

- o Speed V1 (1st stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$" and turned off when the temperature value reaches the value "setpoint-ΔT_{valv}" (or the "setpoint" value if $\Delta T_{1\ heat}$=0). The first speed is also turned off when a higher speed needs to be activated
- o Speed V2 (2nd stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$" and turned off when the temperature value reaches the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$". The second speed is also turned off when the V3 speed needs to be activated
- o Speed V3 (3rd stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$-$\Delta T_{3\ heat}$" and turned off when the temperature value reaches the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$"

With regard to the heating solenoid valve, once the measured temperature is lower than the value "setpoint-ΔT_{valv}", the chronothermostat sends the activation command to the solenoid valve that manages the heating system; the solenoid valve is deactivated when the detected temperature reaches the fixed set-point value. In this way, the heating of the fan coil can also be exploited for irradiation, without any speed being activated.

AIR COOLING



The figure below refers to the control of the speeds of a fan coil with three operating stages for air cooling. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine its activation and deactivation. The thresholds are determined by values set for the various regulation differentials, and can be summarised as follows:

- Speed V1 (1st stage): the speed is turned on when the temperature value is higher than the value "setpoint+$_{\Delta}T_{valv}$+$_{\Delta}T_{1cool}$" and turned off when the temperature value reaches the value "setpoint+$_{\Delta}T_{valv}$" (or the "setpoint" value if $_{\Delta}T_{1\ cool}$=0). The first speed is also turned off when a higher speed needs to be activated
- Speed V2 (2nd stage): the speed is turned on when the temperature value is higher than the value "setpoint+$_{\Delta}T_{valv}$+$_{\Delta}T_{1cool}$+$_{\Delta}T_{2cool}$" and turned off when the temperature value reaches the value "setpoint+$_{\Delta}T_{valv}$+$_{\Delta}T_{1\ cool}$". The second speed is also turned off when the V3 speed needs to be activated
- Speed V3 (3rd stage): the speed is turned on when the temperature value is higher than the value "setpoint+$_{\Delta}T_{valv}$+$_{\Delta}T_{1cond}$+$_{\Delta}T_{2cond}$+$_{\Delta}T_{3cond}$" and turned off when the temperature value reaches the value "setpoint+$_{\Delta}T_{valv}$+$_{\Delta}T_{1cond}$+$_{\Delta}T_{2cond}$"

With regard to the air cooling solenoid valve, once the measured temperature is higher than the value "setpoint+$_{\Delta}T_{valv}$", the chronothermostat sends the activation command to the solenoid valve that manages the air cooling system; the solenoid valve is deactivated when the detected temperature reaches the fixed set-point value. In this way, the air cooling of the fan coil can also be exploited for irradiation, without any speed being activated.

8 "*Burglar system*" menu

The **Burglar system** menu contains the parameters used to enable and configure the control of the panel's basic burglar system functions. The use of a Gewiss control unit connected with the panel via a GW10948 interface is assumed.

The basic structure of the menu is as follows:

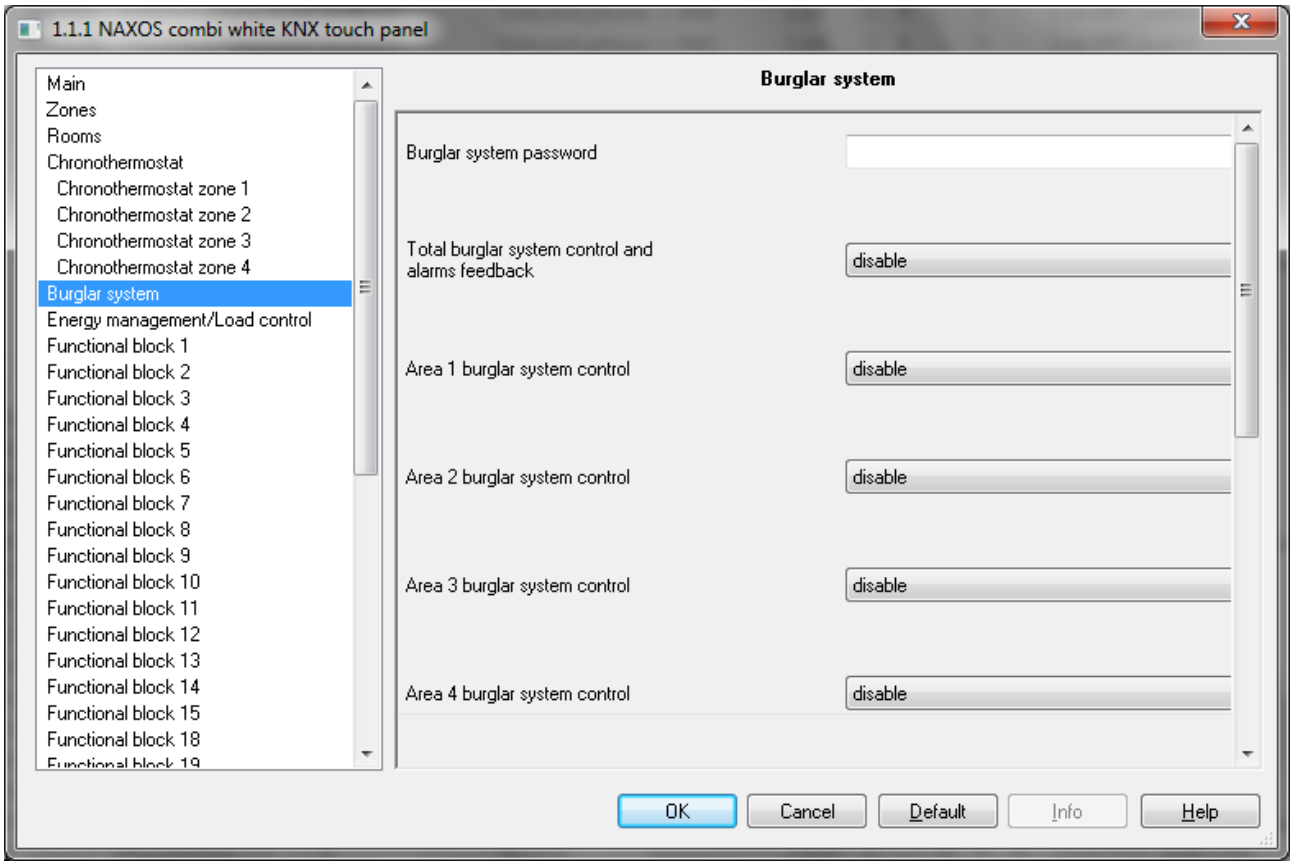


Fig 28: Setting ETS parameters - "Burglar system" section

The parameter **"Burglar system password"** is used to enter up to 8 alphanumeric characters that identify the password associated with the burglar system commands; this password will be requested each time a request is made from the panel to send a command to the burglar system or to access the alarm log. The values that can be set are:

- maximum 8 alphanumeric characters

The parameter **"Total burglar system control and alarms feedback"** is used to enable the control panel to turn total alarm feedback on/off and to receive alarm feedback from the control unit. The values that can be set are:

- **disable** **(default value)**
- enable

selecting **enable**, the configuration menus **Functional block 40** and **Functional block 41** will no longer be displayed, but the following will be displayed:

- the parameter **"Total burglar system off command"**
- the object ***Burglar - Global activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the total on status from the burglar system control unit

- the object ***Burglar - Global activation command*** (Date Point Type: 1.001 DPT_Switch) for sending the total on/off command from the burglar system control unit
- the object ***Burglar - Global activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the enable status from the burglar system control unit
- the object ***Burglar - Global alarm feedback*** (Date Point Type: 1.005 DPT_Alarm) for receiving the global alarm of the control unit
- the object ***Area 1 alarm feedback*** (Date Point Type: 1.005 DPT_Alarm) for receiving the alarm of area 1 of the control unit
- the object ***Area 2 alarm feedback*** (Date Point Type: 1.005 DPT_Alarm) for receiving the alarm of area 2 of the control unit
- the object ***Area 3 alarm feedback*** (Date Point Type: 1.005 DPT_Alarm) for receiving the alarm of area 3 of the control unit
- the object ***Area 4 alarm feedback*** (Date Point Type: 1.005 DPT_Alarm) for receiving the alarm of area 4 of the control unit

The parameter **"Total burglar system off commands"** is used to enable/disable sending the burglar system control unit total off command from the panel via a telegram on the communication object ***Burglar - Global activation command***. The values that can be set are:

- **disable** **(default value)**
- enable

The parameter **"Area 1 burglar system control"** is used to enable the panel to control the switching on/off of sectors 1, 2, 3 and 4 of area 1. The values that can be set are:

- **disable** **(default value)**
- enable

selecting **enable**, the configuration menus **Functional block 38** and **Functional block 39** will no longer be displayed, but the following will be displayed:

- the parameters **"Area name"**, **"Sector 1 name"**, **"Sector 2 name"**, **"Sector 3 name"** and **"Sector 4 name"**
- the object ***Burglar - Area 1 sector 1 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 1 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 1 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 1 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 2 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 2 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 2 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 2 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 3 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 3 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 3 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 3 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 4 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 4 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 4 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 4 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 1 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 1 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 2 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 2 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 3 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 3 belonging to area 1 of the burglar system control unit
- the object ***Burglar - Area 1 sector 4 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 4 belonging to area 1 of the burglar system control unit

The parameter **"Area 2 burglar system control"** is used to enable the panel to control the switching on/off of sectors 1, 2, 3 and 4 of area 2. The values that can be set are:

- **disable** **(default value)**
- enable

selecting **enable**, the configuration menus **Functional block 36** and **Functional block 37** will no longer be displayed, but the following will be displayed:

- the parameters **"Area name"**, **"Sector 1 name"**, **"Sector 2 name"**, **"Sector 3 name"** and **"Sector 4 name"**
- the object ***Burglar - Area 2 sector 1 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 1 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 1 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 1 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 2 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 2 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 2 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 2 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 3 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 3 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 3 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 3 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 4 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 4 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 4 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 4 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 1 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 1 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 2 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 2 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 3 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 3 belonging to area 2 of the burglar system control unit
- the object ***Burglar - Area 2 sector 4 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 4 belonging to area 2 of the burglar system control unit

The parameter **"Area 3 burglar system control"** is used to enable the panel to control the switching on/off of sectors 1, 2, 3 and 4 of area 3. The values that can be set are:

- **disable** **(default value)**
- enable

selecting **enable**, the configuration menus **Functional block 34** and **Functional block 35** will no longer be displayed, but the following will be displayed:

- the parameters **"Area name"**, **"Sector 1 name"** , **"Sector 2 name"** , **"Sector 3 name"** and **"Sector 4 name"**
- the object ***Burglar - Area 3 sector 1 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 1 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 1 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 1 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 2 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 2 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 2 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 2 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 3 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 3 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 3 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 3 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 4 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 4 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 4 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 4 belonging to area 3 of the burglar system control unit

- the object ***Burglar - Area 3 sector 1 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 1 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 2 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 2 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 3 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 3 belonging to area 3 of the burglar system control unit
- the object ***Burglar - Area 3 sector 4 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 4 belonging to area 3 of the burglar system control unit

The parameter **"Area 4 burglar system control"** is used to enable the panel to control the switching on/off of sectors 1, 2, 3 and 4 of area 4. The values that can be set are:

- **disable (default value)**
- enable

selecting **enable**, the configuration menus **Functional block 32** and **Functional block 33** will no longer be displayed, but the following will be displayed:

- the parameters **"Area name"**, **"Sector 1 name"** , **"Sector 2 name"** , **"Sector 3 name"** and **"Sector 4 name"**
- the object ***Burglar - Area 4 sector 1 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 1 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 1 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 1 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 2 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 2 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 2 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 2 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 3 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 3 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 3 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 3 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 4 activation status*** (Date Point Type: 1.001 DPT_Switch) for receiving the on status of sector 4 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 4 command*** (Date Point Type: 1.001 DPT_Switch) for receiving the on/off commands for sector 4 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 1 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 1 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 2 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 2 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 3 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 3 belonging to area 4 of the burglar system control unit
- the object ***Burglar - Area 4 sector 4 activation enabling*** (Date Point Type: 1.003 DPT_Enable) for receiving the on enable status of sector 4 belonging to area 4 of the burglar system control unit

The parameter **"Area name"** is used to enter the name you want to associate with the graphic element that represents the area (1, 2, 3 or 4 depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

The parameter **"Sector name i"** is used to enter the name you want to associate with the graphic element that represents the sector i (1..4) of a certain area (1, 2, 3 or 4 depending on the header to which the parameter refers) in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

9 “*Energy management/Load control*” menu

The **Energy management/Load control** menu contains the parameters used to enable and configure the communication objects used to receive the consumption relative to electricity, water and gas, which will then be recorded and displayed by the panel or used in the load control function.

The basic structure of the menu is as follows:

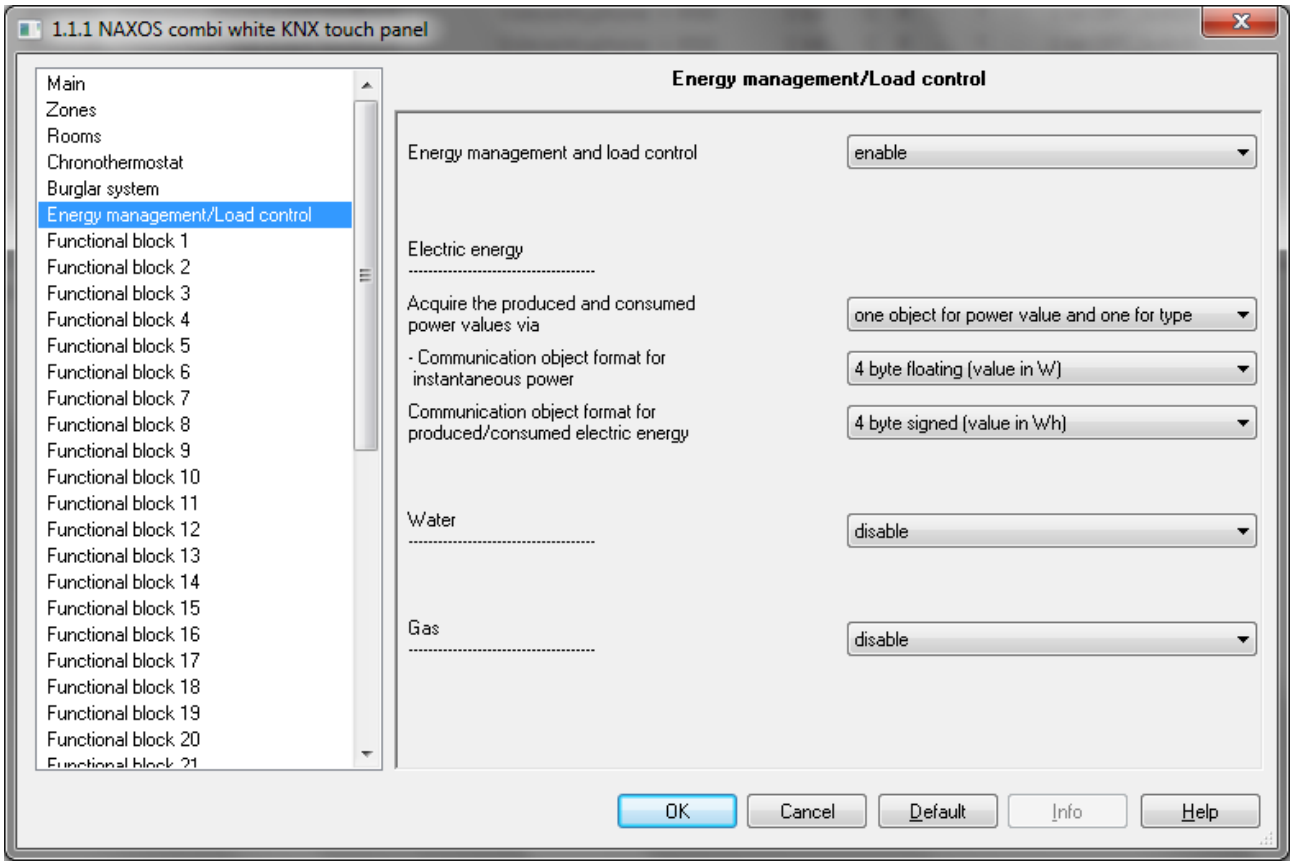


Fig 29: Setting ETS parameters - “Energy management/Load control” section

The parameter **“Energy management and load control”** is used to enable the “Energy management ” and “Load control” functions, displaying the communication objects and main parameters of the functions. The values that can be set are:

- **disable** **(default value)**
- enable

by selecting **enable**, the configuration menu **Functional block 42** will no longer be displayed whereas the parameters **“Acquire the produced and consumed power values via”** and **“Communication object format for produced/consumed electric energy”** are displayed in the Electricity section, together with the **“Water”** and **“Gas”** parameters.

In order to interface with a greater number of Energy meters available on the market, the parameter **“Acquire the produced and consumed power values via”** can be used to define the type of communication objects used to receive instantaneous consumed/produced power. The values that can be set are:

- one object only for consumed power
- one signed object: + consumed | - produced
- one signed object: + produced | - consumed
- **one object for power value and one for type** **(default value)**
- two independent objects

setting **one object only for consumed power** displays the parameter **"Communication object format for instantaneous consumed power"** and the communication object ***Energy management - Active power consumed*** via which the instantaneous consumed power value is received.
Setting **one signed object: + consumed | - produced**, displays the parameter **"Communication object format for instantaneous consumed/produced power"** and the communication object ***Energy management - Active signed power*** via which the instantaneous consumed power value (if the received value is greater 0) and produced power value (if the received value is lower than 0) is received.
Setting **one signed object: + produced | - consumed**, displays the parameter **"Communication object format for instantaneous consumed/produced power"** and the communication object ***Energy management - Active signed power*** via which the instantaneous consumed power value (if the received value is lower than 0) and produced power value (if the received value is greater than 0) is received.
Setting **one object for power value and one for type** displays the parameter **"Communication object format for instantaneous power"** and the communication objects ***Energy management - Active power*** and ***Energy management - Active power/Load type*** (Date Point Type: Non standard) used respectively to receive the absolute instantaneous power value and the type (produced or consumed). The object is coded as follows:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not used	Not used.	Not used	Not used	Consumed power	Produced power	Inductive load	Capacitive load

Setting **two independent objects** displays the parameter **"Communication objects format for instantaneous produced/consumed power"** and the communication objects ***Energy management - Active power consumed*** and ***Energy management - Active power produced*** used respectively to receive the instantaneous consumed power value and the instantaneous produced power value.

The parameter **"Communication object format for instantaneous consumed power"** which is displayed if the power is acquired via "one object only for consumed power ", is used to define the format and coding of the BUS telegrams that will be received by the panel via the communication object ***Energy management - Active power consumed***. the possible values are:

- 2 bytes floating
  with this setting, the format of the above cited object is *9.024 DPT_Power [kW]*
- **4 bytes floating** **(default value)**
  with this setting, the format of the above cited object is *14.056 DPT_Value_Power [W]*

The parameter **"Communication object format for instantaneous consumed/produced power"**, which is displayed if the power is acquired via "one signed object: + consumed | - produced" or "one signed object: + produced| - consumed", is used to define the format and coding of the BUS telegrams that will be received by the panel via the communication object ***Energy management - Active signed power***. The possible values are:

- 2 bytes floating
  with this setting, the format of the above cited object is *9.024 DPT_Power [kW]*
- **4 bytes floating** **(default value)**
  with this setting, the format of the above cited object is *14.056 DPT_Value_Power [W]*

The parameter **"Communication object format for instantaneous power"** which is displayed if the power is acquired via "one object for the power value and one for the sign", is used to define the format and coding of the BUS telegrams that will be received by the panel via the communication object ***Energy management - Active power***. The possible values are:

- 2 bytes floating
  with this setting, the format of the above cited object is *9.024 DPT_Power [kW]*
- **4 bytes floating** **(default value)**
  with this setting, the format of the above cited object is *14.056 DPT_Value_Power [W]*

The parameter **"Communication objects format for instantaneous produced/consumed power"** which is displayed if the power is acquired via "two independent objects", is used to define the format and coding of the BUS telegrams that will be received by the panel via the communication objects ***Energy management - Active power consumed*** and ***Energy management - Active power produced***. The possible values are:

- 2 bytes floating
  with this setting, the format of the above cited object is *9.024 DPT_Power [kW]*
- **4 bytes floating (default value)**
  with this setting, the format of the above cited object is *14.056 DPT_Value_Power [W]*

The parameter **"Communication object format for produced/consumed electric energy"** is used to define the format and coding of the BUS telegrams that will be received by the panel via the communication objects ***Energy management - Active energy consumed*** and ***Energy management - Active energy produced***. The possible values are:

- **4 bytes signed (value in Wh) (default value)**
  with this setting, the format of the above cited objects is *13.010 DPT_ActiveEnergy*
- 4 bytes signed (value in kWh)
  with this setting, the format of the above cited objects is *13.013 DPT_ActiveEnergy_kWh*

The parameter **"Water"** is used to enable the communication object for calculating the water volume. The values that can be set are:

- **disable (default value)**
- enable

selecting **enable** displays the communication object ***Energy management - Water counter*** (Date Point Type: 14.076 DPT_Value_Volume) used to receive the volume of consumed water.

The parameter **"Gas"** is used to enable the communication object for calculating the gas volume. The values that can be set are:

- **disable (default value)**
- enable

selecting **enable** displays the communication object ***Energy management - Gas counter*** (Date Point Type: 14.076 DPT_Value_Volume) used to receive the volume of consumed gas.

NOTE: in order to be assigned to the selected load control band of the panel, this type of elements must be associated to one of the following functions: "1 light", "2 lights", "3 lights", "1 on/off actuator", "2 on/off actuators" or "3 on/off actuators".

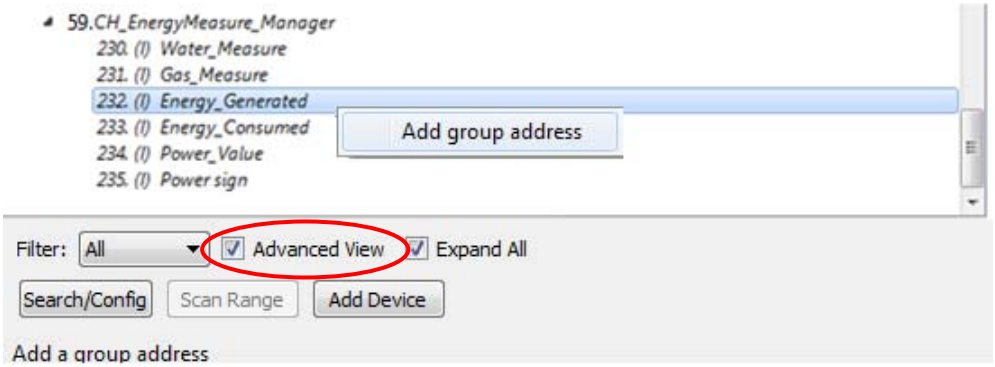
9.1 Programming in Easy mode

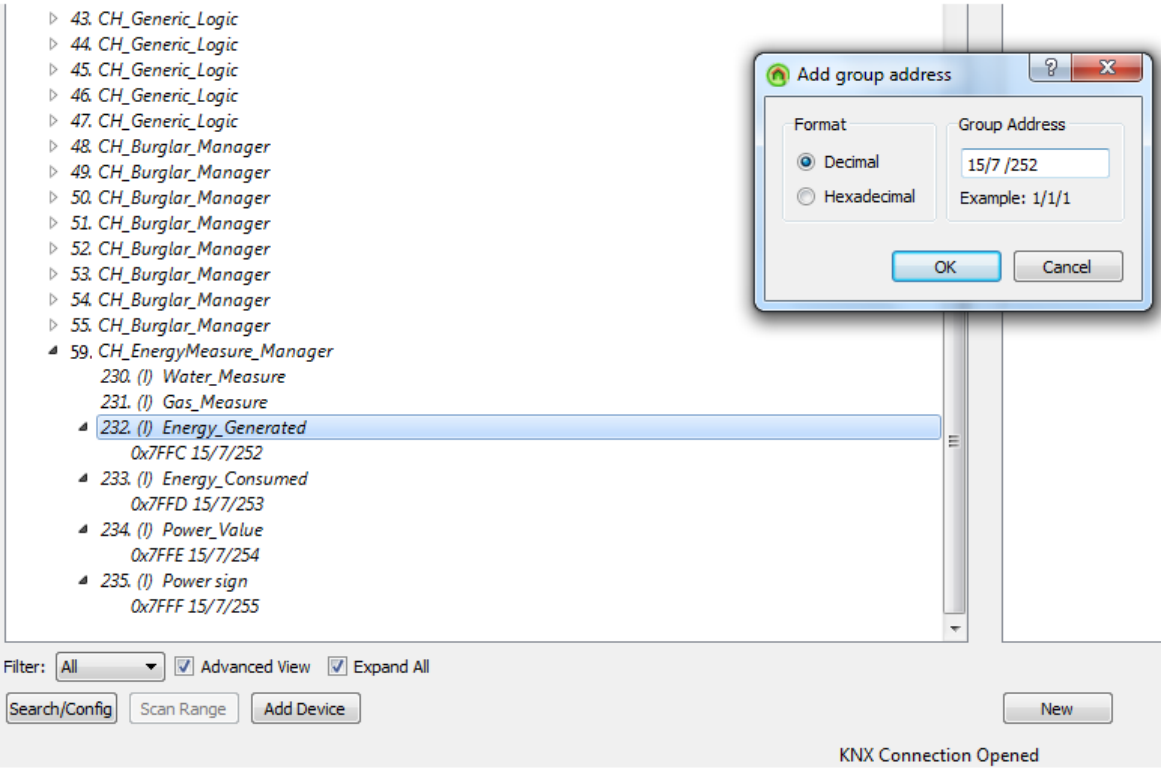
If the panels would be commissioned in Easy mode and therefore by the Easycontroller software, it is necessary to know that GW90876 KNX interface for Energy meter isn't an Easy device, the interface would not be recognized by the Easycontroller software and so it is important to know the group addresses pre-configured on the GW90876 interface in order to activate the Energy management and the Load control functions.
The group addresses pre-configured on the interface GW90876 are the followings:

- 15/7/252: for the Energy generated (KWh)
- 15/7/253: for the Energy consumed (KWh)
- 15/7/254: for the Power value (W)
- 15/7/255: for the Power sign

9.1.1 Configuration using Easycontroller software

The program of Energy mangement and Load control functions are made combing the above group addresses to the Naxos's channels.
Below two pictures and some notes in manner to clarify how to do it, paying attention that it's ***possible only by the on-line mode*** and not by the off-line.
The section 59 of the panels is dedicated to the Energy management and the Load control; activating the Advanced View as shown on the images below, it is possible to add a group address selecting the channel interested and through a right-click of the mosue.

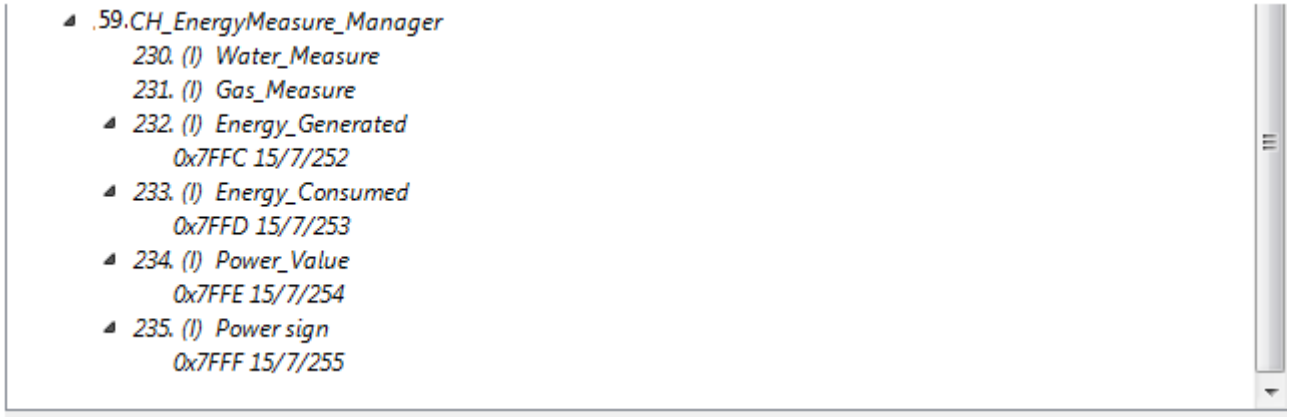




Following the associations between channels and group addresses:

- Channel_232 at 15/7/252
- Channel_233 at 15/7/253
- Channel_234 at 15/7/254
- Channel_235 at 15/7/255

Once the associations have been completed, Easycontroller software displays them as shown on the image below and the Naxos panels will receive the measurements by the interface GW90876.



10 "*Logics and conditions*" menu

The **Logics and conditions** menu contains the only parameters that can enable and configure the operation of each of the 16 available logic blocks.

The basic structure of the menu is as follows:

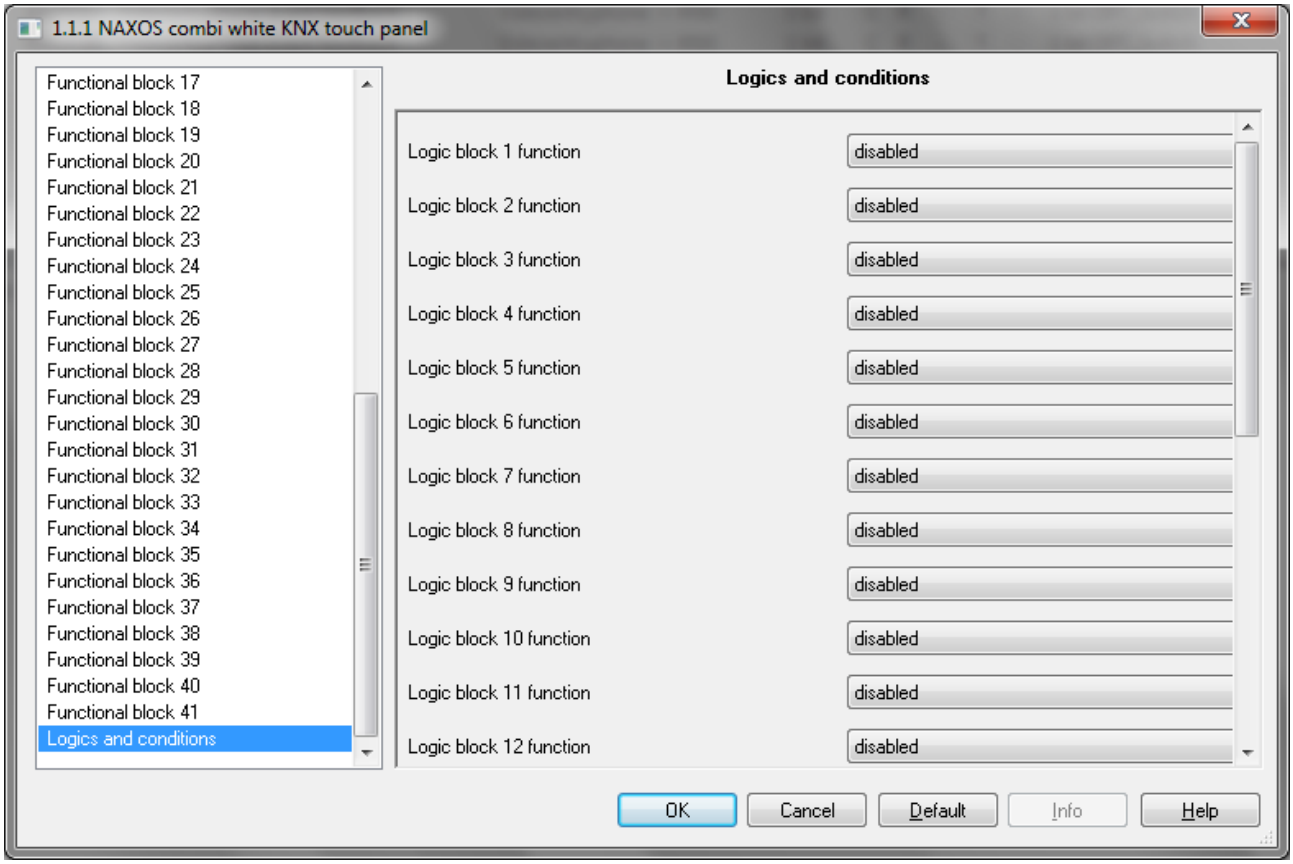


Fig 30: Setting ETS parameters - "Logics and conditions" section

The parameters **"Logic block 1 function"**, …**"Logic block 16 function"** are used to select the block operation and display and configure all the operating parameters for the relative logic blocks grouped in the menus **Logic block 1**, .. **Logic block 16**. The values that can be set for these parameters are:

- **disabled (default value)**
- enable logic operation
- enable **comparison**

setting the value **enable logic operation**, the operation of the associated block is that of the logic operator (AND, OR etc.) whereas by selecting **enable comparison** the operation will be that of a comparator (">", "<", "=" etc..); the parameter that will be displayed in the corresponding configuration menu also depend on the type of associated logic function.

10.1 "Logic block x - logic operation" menu

For the sake of simplicity, the items that make up the menus **Logic block 1**, … **Logic block 16** will be described only once in the following chapters (in reference to the generic menu **Logic block x**) as all these menus have the same items.
If configured as a logic operation, the **Logic block x** menu contains the parameters used to configure the logic operation to be executed, also defining the involved objects.

The graphic element that represents this function, inside the navigation tree belongs to the "Programs→Logics" category.

The basic structure of the menu is as follows:

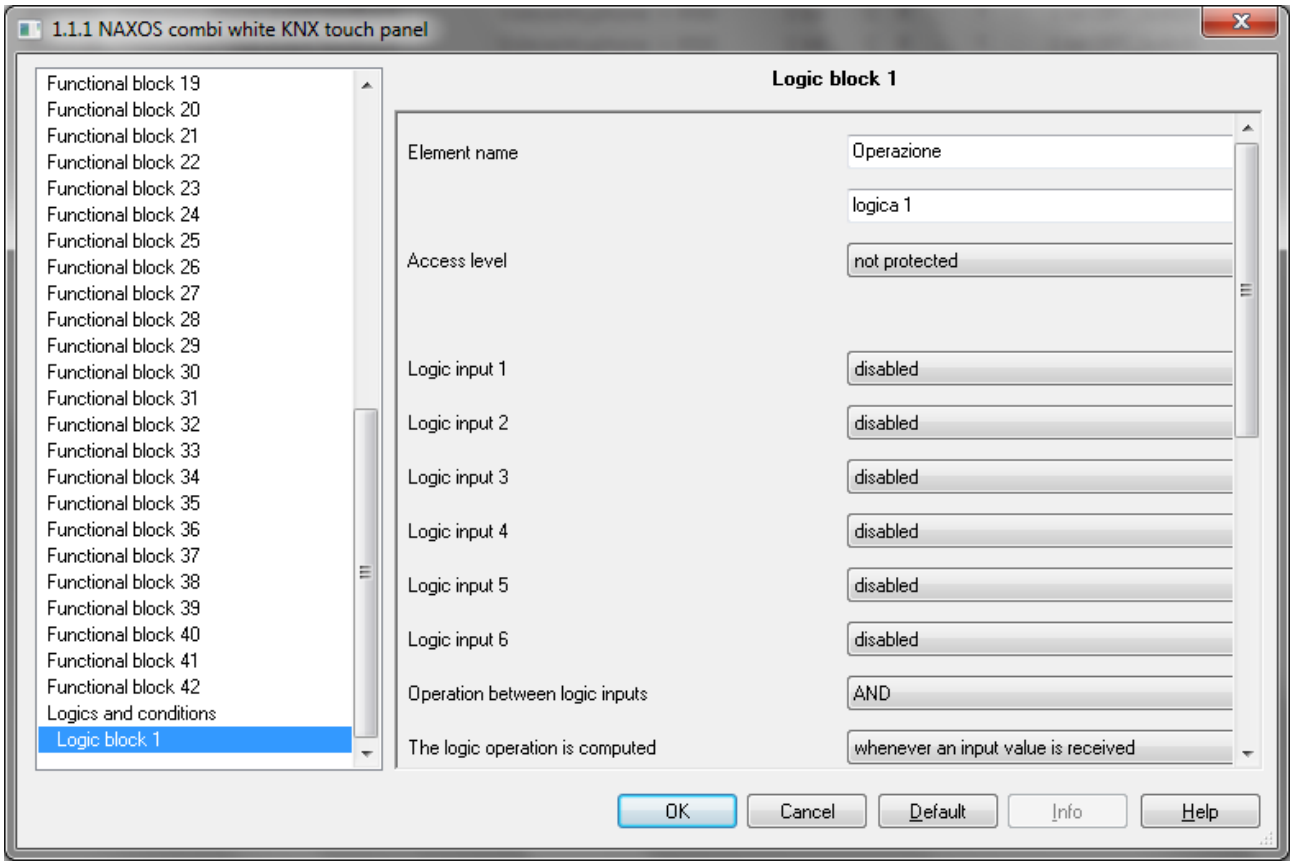


Fig 31: Setting ETS parameters - "Functional block x - logic operation" section

The parameter **" Element name"** is used to enter the name you want to associate with the logic operation x element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for activating/deactivating the logic operation via the graphic element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

The logic operation can have a maximum of 6 inputs; via the parameters **"Logic input 1"**, … **"Logic input 6"** it is possible to enable and define the type of input to be used in the logic operation. The values that can be set for the parameters listed above are:

- **disabled (default value)**
- communication object
- logic function output

by selecting **communication object**, it is the value of a communication object present in the project (there are no communication objects dedicated to implementing logic functions) that is used as a logic input; this setting displays the parameter **"Communication object associated with logic input 1"** (or **"Communication object associated with logic input 2"** … **"Communication object associated with logic input 6"**).
By selecting **logic function output**, it is the result of another programmed logic function that is used as the logic input; this setting displays the parameter **"Logic output associated with logic input 1"** (or **"Logic output associated with logic input 2"** … **"Logic output associated with logic input 6"**).

The parameter **"Communication object associated with logic input 1"** … **"Communication object associated with logic input 6"** is used to select the index of the communication object to be used as the input for the logic operation. the values that can be set are:

- from **0 (default value)** to 251 with step of 1

The parameter **"Logic output associated with logic input 1"** … **"Logic output associated with logic input 6"** is used to select the index of the logic function whose output is to be used as the input for the logic operation; if the logic function selected was not configured or was not configured correctly, the logic operation has been configured incorrectly and will not be implemented/displayed by the graphic interface. the values that can be set are:

- **logic block 1 output (default value)**
- logic block 2 output
  …
- logic block 16 output

The logic operation to be executed between the logic inputs is defined via the parameter **"Operation between logic inputs"** which can have the following values:

- **AND (default value)**
- OR
- NAND
- NOR
- XOR
- XNOR

It is possible to determine the condition that generates the calculation of the logic operation selected via the parameter; the values that can be set are:

- **whenever an input value is received (default value)**
  the logic is calculated each time an input value is received, independently of the fact if the new value is equal to or different than the previous value
- if at least one input value changes status
  the logic is calculated each time an input value is received only if the new value is equal to or different than the previous value
- periodically
  the logic is calculated at fixed time ranges, independently of the fact if new input values were received

Selecting **whenever an input value is received** or **if at least one input value changes status** displays the parameter **"Minimum interval between two logic evaluations"**.
Selecting **periodically** displays the parameter **"Calculation period"**.

Given the possibility to concatenate the logic functions (setting for example the output of a logic block as the input for a new logic block) situations can be created where the logic is calculated continuously, effectively

blocking panel operation. To avoid this malfunction, the minimum interval between two subsequent logic evaluations is defined via the parameter **"Minimum interval between two logic evaluations"**, so that even if the condition for calculating the logic operation takes place, it is only calculated if at least a period of time greater than the minimum set interval has passed. The values that can be set are:

- from **1 second (default value)** to 255 seconds with step of 1

The parameter **"Calculation period"** is used to display the time range with which the logic is calculated. The values that can be set are:

- from **1 second (default value)** to 255 seconds with step of 1

It is possible to refuse the value from the logic inputs, independently of the fact if they are communication objects or logic function outputs, via the parameters **"NOT operation for logic input 1"** … **"NOT operation for logic input 6"** (which is displayed depending on the number of enabled logic inputs), which can have the following values:

- **disabled (default value)**
- enable

The result of the logic operation can be transmitted on the KNX BUS via the communication objects present in the project (there are no communication objects dedicated to implementing logic functions). The conditions for sending the telegram associated with the result of the logic operation is defined via the parameter **"Generate the result of the logic operation"**, which can have the following values:

- **whenever the logic is evaluated (default value)**
  the telegrams are sent each time the logic is calculated, independently of the fact that the result of the new calculation is equal to or different than the previous result
- only if the result changes
  the telegrams are only sent if the result of the new calculation is different than the previous result

The result of the logic operation can be "true" or "false"; for each of these values, it is possible to define if a feedback telegram should be sent or not via a communication object. The parameter **"If the result of the logic operation is TRUE"** is used to activate the sending of the result feedback "true" whereas the parameter **"If the result of the logic operation is FALSE"** the sending of the result feedback "false"; The possible values for these parameters are:

- **do not send bus commands (default value)**
- send bus command

selecting **send bus command** displays the parameters **"Communication object index"**, **"Object size"**, **"Object format"** and **"Value"**.

The parameter **"Communication object index"** is used to select the index of the communication object to be used to signal the "true" or "false" result of the logic operation, depending on the result to which the parameter refers. the values that can be set are:

- from **0 (default value)** to 251 with step of 1

The parameter **"Object size"** is used to set the size of the communication object to be used to signal the "true" or "false" result of the logic operation, depending on the result to which the parameter refers. The values that can be set are:

- **1 bit (default value)**
- 2 bits
- 4 bits
- 1 byte
- 2 bytes
- 3 bytes
- 4 bytes

Based on the value set for this item, the values that can be set for the parameter **" Object format"** will change as a result.
The parameter **"Object format"** is used to set the size of the communication object to be used to signal the "true" or "false" result of the logic operation, depending on the result to which the parameter refers. The values that can be set are:

- If the object size is **1 bit**:

  - **on/off** **(default value)**
  - boolean
  - shutter up/down
  - increase/decrease
  - heating/air cooling
  - enable/disable
  - alarm/no alarm
  - start/Stop
  - occupied/free
  - open/closed

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the logic operation is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

  - If the object format is on/off:
    - **off** **(default value if the result is false)**
    - **on** **(default value if the result is true)**
  - If the format object is boolean:
    - **false** **(default value if the result is false)**
    - **true** **(default value if the result is true)**
  - If the format object is shutter up/down:
    - **up** **(default value if the result is false)**
    - **down** **(default value if the result is true)**
  - If the format object is increase/decrease:
    - **decrease** **(default value if the result is false)**
    - **increase** **(default value if the result is true)**
  - If the format object is heating/cooling:
    - **cooling** **(default value if the result is false)**
    - **heating** **(default value if the result is true)**
  - If the format object is enable/disable:
    - **disable** **(default value if the result is false)**
    - **enable** **(default value if the result is true)**
  - If the format object is alarm/no alarm:
    - **no alarm** **(default value if the result is false)**
    - **alarm** **(default value if the result is true)**
  - If the format object is start/stop:
    - **stop** **(default value if the result is false)**
    - **start** **(default value if the result is true)**
  - If the format object is occupied/free:
    - **free** **(default value if the result is false)**
    - **occupied** **(default value if the result is true)**
  - If the format object is open/closed:
    - **open** **(default value if the result is false)**
    - **closed** **(default value if the result is true)**

- If the object size is **2 bits**:

- **on/off forced positioning (default value)**
- up/down forced positioning
  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the logic operation is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

- If the object format is on/off forced positioning:
  - **deactivate forcing** **(default value)**
  - active off forcing
  - active on forcing
- If the format object is up/down forced positioning:
  - **deactivate forcing** **(default value)**
  - active up forcing
  - active down forcing

- If the object size is **4 bits**:

  - **dimmer step** **(default value)**
  - shutter step

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the logic operation is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

  - If the format object is dimmer step:
    - **stop regulation** **(default value)**
    - decrease 100%
    - decrease 50%
    - decrease 25%
    - decrease 12.5%
    - decrease 6.25%
    - decrease 3.125%
    - decrease 1.56%
    - increase 100%
    - increase 50%
    - increase 25%
    - increase 12.5%
    - increase 6.25%
    - increase 3.125%
    - increase 1.56%
  - If the format object is shutter step:
    - **stop regulation** **(default value)**
    - raise 100%
    - raise 50%
    - raise 25%
    - raise 12.5%
    - raise 6.25%
    - raise 3.125%
    - raise 1.56%
    - lower 100%
    - lower 50%
    - lower 25%
    - lower 12.5%
    - lower 6.25%
    - lower 3.125%
    - lower 1.56%

- If the object size is **1 byte**:

  - **unsigned** **(default value)**
  - signed
  - percentage
  - two's complement percentage
  - HVAC mode
  - scene
  - angular width (degrees)

Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the logic operation is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

- If the format object is unsigned:
  - from **0 (default value)** to 255, with steps of 1
- If the format object is signed:
  - from -128 to 127 with step of 1, **0 (default value)**
- If the format object is percentage:
  - from **0% (default value)** to 100% with step of 5%
- If the format object is two's complement percentage
  - from -128 to 127 with step of 1, **0 (default value)**
- If the format object is HVAC mode:
  - **auto** **(default value)**
  - comfort
  - precomfort
  - economy
  - off (building protection)
- If the format object is scene:
  - **execute scene 0** **(default value)**
  - execute scene 1
    …
  - execute scene 63
  - learn scene 0
  - learn scene 1
    …
  - learn scene 63
- If the format object is angular width (degrees):
  - from **0° (default value)** to 360°

- If the object size is **2 bytes**:

  - **unsigned** **(default value)**
  - signed
  - time in seconds
  - time in minutes
  - time in hours
  - floating temperature °C
  - floating temperature °K
  - floating temperature °F
  - floating brightness
  - floating humidity
  - floating solar intensity
  - floating voltage
  - floating current
  - floating power
  - floating wind speed

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the logic operation is "true" or "false", depending on the result to which the parameter refers; the "floating" values could differ from what is actually sent due to conversion. The values that can be set are:

  - If the object format is unsigned, time in seconds, time in minutes or time in hours:
    - from **0 (default value)** to 65535, with steps of 1
  - If the format object is signed:
    - from **-**32768 to 32767 with step of 1, **0 (default value)**
  - the format object is floating °C:
    - from -273 to 670760 with step of 1 for the whole part, **0 (default value)**
    - from **0 (default value)** to 99 with step of 1 for the decimal part

- If the format of the object is floating °K, floating solar intensity, floating voltage, floating current and floating power:
  - from -670760 to 670760 with step of 1 for the whole part, **0 (default value)**
  - from **0 (default value)** to 99 with step of 1 for the decimal part
- the format object is floating °F:
  - from -459 to 670760 with step of 1 for the whole part, **0 (default value)**
  - from **0 (default value)** to 99 with step of 1 for the decimal part
- If the object format is floating brightness, floating humidity or floating wind speed:
  - from **0 (default value)** to 670760 with step of 1 for the whole part
  - from **0 (default value)** to 99 with step of 1 for the decimal part

- If the object size is **3 bytes**:

  - **RGB** **34 (default value)**

  this setting displays the parameters **"RED component value"**, **"GREEN component value"** and **"BLUE component value"** which are used to set the value of each component to be sent when the result of the logic operation is "true" or "false", depending on the result to which the parameter refers. The values that can be set are:

  - from **0 (default value)** to 255, with steps of 1

- If the object size is **4 bytes**:

  - **unsigned** **1 (default value)**
  - signed
  - signed active power
  - signed reactive power
  - signed apparent power
  - signed active power (kWh)
  - signed reactive power (kVARh)
  - signed apparent power (kVAh)
  - floating power
  - floating power factor
  - floating speed
  - floating volume

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the logic operation is "true" or "false", depending on the result to which the parameter refers; the "floating" values could differ from what is actually sent due to conversion. The values that can be set are:

  - If the format object is unsigned:
    - from **0 (default value)** to 4294967295, with steps of 1
  - If the object format is signed, with signed active power, with signed reactive power, with signed apparent power, signed active power(kWh), with signed reactive power(kVARh) or with signed apparent power (kVAh):
    - from -2147483648 to 2147483647 with steps of 1, **0 (default value)**
  - If the object format is floating power, floating power factor, floating speed or floating volume:
    - from -2147483648 to 2147483647 with step of 1 for the whole part, **0 (default value)**
    - from **0 (default value)** to 99 with step of 1 for the decimal part

In addition to the possibility of sending commands, it is possible to associate the activation of the buzzer present on the device to the result of the logic operation via the parameter **"Acoustic signal (buzzer)"**, which permits both enabling as well as the possible activation conditions; the values that can be set are:

- **disabled** **(default value)**
- enabled if the event is TRUE
- enabled if the event is FALSE
- enabled for both events

10.2 "Logic block x - comparison" menu

For the sake of simplicity, the items that make up the menus **Logic block 1** … **Logic block 16** will be described only once in the following chapters (in reference to the generic menu **Logic block x**) as all these menus have the same items.
If configured as a comparison, the **Logic block x** menu contains the parameters used to configure the comparison to be executed, also defining the involved objects.

The graphic element that represents this function, inside the navigation tree belongs to the "Programs→Logics" category.

The basic structure of the menu is as follows:

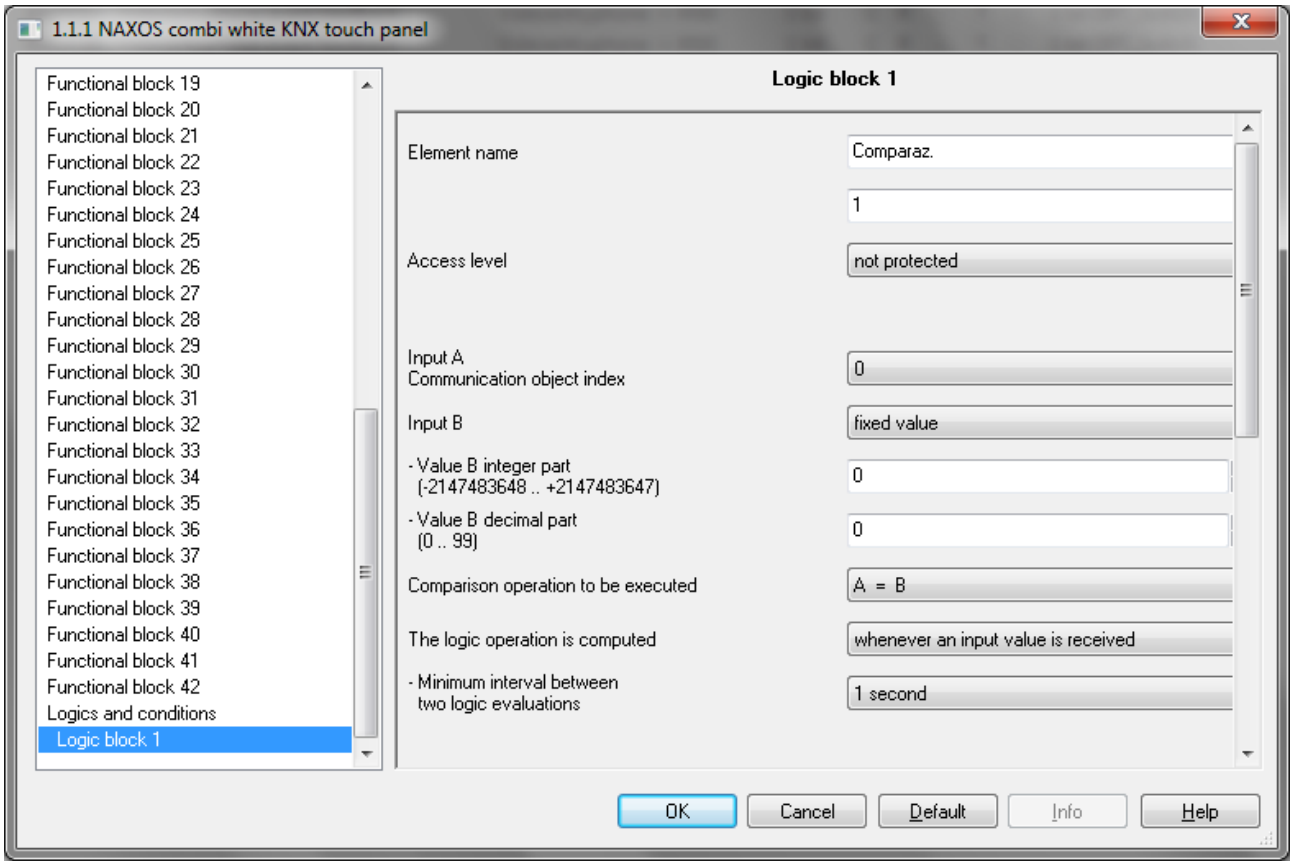


Fig 32: Setting ETS parameters - "Logic block x - comparison" section

The parameter **" Element name"** is used to enter the name you want to associate with the logic comparison x element in the project; the name consists of two rows, as the element is represented on a graphical level inside tiles with defined dimensions. When the element name is instead represented on a single row, then a space is automatically inserted between the string inserted in the first row and the one inserted in the second. The possible values are:

- maximum 10 alphanumeric characters per row

Via the parameter **"Access level"** it is possible to define the rights required for activating/deactivating the comparison via the graphic element; even if protected, the element is still visible with its status information. The possible values are:

- **not protected** **(default value)**
- protection level 1 .. 3

In the comparison, one of the two logic inputs is always the value received via a communication object present in the project (there are no communication objects dedicated to implementing logic functions); the parameter **"Input A Communication object index"** is used to select the index of the communication object to be used as the first input for the comparison; if the object selected as an input is an object dedicated to transmitting telegrams and not for receiving, the logic function has been configured incorrectly and will not be implemented/displayed by the graphic interface. the values that can be set are:

- from **0 (default value)** to 251 with step of 1

The comparison can be made between two values received from the BUS or between one value received from the BUS and a fixed value defined via ETS; the parameter **"Input B"** is used to define the nature of the second input for the comparison. The values that can be set are:

- **communication object** **0 (default value)**
- fixed value 1

selecting the value **communication object** displays the parameter **"Input B communication object index"** whereas selecting **fixed value** to display the parameters **"Value B integer part (-2147483648 .. +2147483647)"** and **"Value B decimal part (0 .. 99)"**.

the parameter **"Input B communication object index"** is used to select the index of the communication object to be used as the second input for the comparison. the values that can be set are:

- from **0 (default value)** to 251 with step of 1

The parameter **"Value B integer part (-2147483648 .. +2147483647)"** is used to set the whole part of the value B with which the first input (A) must be compared. the values that can be set are:

- from -2147483648 to 2147483647with step of 1, **0 (default value)**

The parameter **"Value B decimal part (0 .. 99)"** is used to set the decimal part of the value B with which the first input (A) must be compared. the values that can be set are:

- from **0 (default value)** to 99 with step of 1

The comparison operation to be executed between the input A and input B is defined via the parameter **"Operation between logic inputs"** which can have the following values:

- **A = B** **(default value)**
- A != B
- A > B
- A >= B
- A < B
- A <= B

It is possible to determine the condition that generates the comparison selected via the parameter; the values that can be set are:

- **whenever an input value is received** **(default value)**
  the comparison is carried out each time an input value is received, independently of the fact if the new value is equal to or different than the previous value
- if at least one input value changes status
  the comparison is carried out each time an input value is received only if the new value is equal to or different than the previous value

- periodically
  the comparison is carried at fixed time ranges, independently of the fact if new input values were received

Selecting **whenever an input value is received** or **if at least one input value changes status** displays the parameter **"Minimum interval between two logic evaluations"**.
Selecting **periodically** displays the parameter **"Calculation period"**.

Given the possibility to concatenate the logic functions (setting for example the output of a logic block as the input for a new logic block) situations can be created where the logic is calculated continuously, effectively blocking panel operation. To avoid this malfunction, the minimum interval between two subsequent logic evaluations is defined via the parameter **"Minimum interval between two logic evaluations"**, so that even if condition for calculating the comparison takes place, it is only calculated if at least a period of time greater than the minimum set interval has passed. The values that can be set are:

- from **1 second (default value)** to 255 seconds with step of 1

The parameter **"Calculation period"** is used to display the time range with which the comparison is carried out. The values that can be set are:

- from **1 second (default value)** to 255 seconds with step of 1

The result of the comparison can be transmitted on the KNX BUS via the communication objects present in the project (there are no communication objects dedicated to implementing logic functions). The conditions for sending the telegram associated with the result of the comparison is defined via the parameter **"Generate the result of the comparison"**, which can have the following values:

- **each time the logic is evaluated (default value)**
  the telegrams are sent each time the comparison is carried out, independently of the fact that the result of the new calculation is equal to or different than the previous result
- only if the outcome changes
  the telegrams are only sent if the result of the new calculation is different than the previous result

The result of the comparison can be "true" or "false"; for each of these values, it is possible to define if a feedback telegram should be sent or not via a communication object. The parameter **"If the result of the comparison is TRUE"** is used to activate the sending of the result feedback "true" whereas the parameter **"If the result of the comparison is FALSE "** the sending of the result feedback "false"; The possible values for these parameters are:

- **do not send BUS commands (default value)**
- send BUS command

selecting **send BUS command** displays the parameters **"Communication object index"**, **"Object size"**, **"Object format"** and **"Value"**.

The parameter **"Communication object index"** is used to select the index of the communication object to be used to signal the "true" or "false" result of the comparison, depending on the result to which the parameter refers. the values that can be set are:

- from **0 (default value)** to 251 with step of 1

The parameter **"Object size"** is used to set the size of the communication object to be used to signal the "true" or "false" result of the comparison, depending on the result to which the parameter refers. The values that can be set are:

- **1 bit (default value)**
- 2 bits
- 4 bits
- 1 byte
- 2 bytes
- 3 bytes
- 4 bytes

Based on the value set for this item, the values that can be set for the parameter **" Object format"** will change as a result.
The parameter **"Object format"** is used to set the size of the communication object to be used to signal the "true" or "false" result of the comparison, depending on the result to which the parameter refers. The values that can be set are:

- If the object size is **1 bit**:

  - **on/off** **(default value)**
  - boolean
  - shutter up/down
  - increase/decrease
  - heating/air cooling
  - enable/disable
  - alarm/no alarm
  - start/stop
  - occupied/free
  - open/closed

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the comparison is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

  - o If the object format is on/off:
    - **off** **(default value if the result is false)**
    - **on** **(default value if the result is true)**
  - o If the format object is boolean:
    - **false** **(default value if the result is false)**
    - **true** **(default value if the result is true)**
  - o If the format object is shutter up/down:
    - **up** **(default value if the result is false)**
    - **down** **(default value if the result is true)**
  - o If the format object is increase/decrease:
    - **decrease** **(default value if the result is false)**
    - **increase** **(default value if the result is true)**
  - o If the format object is heating/cooling:
    - **cooling** **(default value if the result is false)**
    - **heating** **(default value if the result is true)**
  - o If the format object is enable/disable:
    - **disable** **(default value if the result is false)**
    - **enable** **(default value if the result is true)**
  - o If the format object is alarm/no alarm:
    - **no alarm** **(default value if the result is false)**
    - **alarm** **(default value if the result is true)**
  - o If the format object is start/stop:
    - **stop** **(default value if the result is false)**
    - **start** **(default value if the result is true)**
  - o If the format object is occupied/free:
    - **free** **(default value if the result is false)**
    - **occupied** **(default value if the result is true)**
  - o If the format object is open/closed:
    - **open** **(default value if the result is false)**
    - **closed** **(default value if the result is true)**

- If the object size is **2 bits**:

  - **on/off forced positioning (default value)**
  - up/down forced positioning

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the comparison is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

  - o If the object format is on/off forced positioning:
    - **deactivate forcing** **(default value)**
    - active off forcing

  - active on forcing
  - If the format object is up/down forced positioning:
    - **deactivate forcing** **(default value)**
    - active up forcing
    - active down forcing

- If the object size is **4 bits**:

  - **dimmer step** **(default value)**
  - shutter step

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the comparison is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

  - If the format object is dimmer step:
    - **stop regulation** **(default value)**
    - decrease 100%
    - decrease 50%
    - decrease 25%
    - decrease 12.5%
    - decrease 6.25%
    - decrease 3.125%
    - decrease 1.56%
    - increase 100%
    - increase 50%
    - increase 25%
    - increase 12.5%
    - increase 6.25%
    - increase 3.125%
    - increase 1.56%
  - If the format object is shutter step:
    - **stop regulation** **(default value)**
    - raise 100%
    - raise 50%
    - raise 25%
    - raise 12.5%
    - raise 6.25%
    - raise 3.125%
    - raise 1.56%
    - lower 100%
    - lower 50%
    - lower 25%
    - lower 12.5%
    - lower 6.25%
    - lower 3.125%
    - lower 1.56%

- If the object size is **1 byte**:

  - **unsigned** **(default value)**
  - signed
  - percentage
  - two's complement percentage
  - HVAC mode
  - scene
  - angular width (degrees)

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the comparison is "true" or "false", depending on the result to which the parameter refers; the values that can be set are:

- If the format object is unsigned:
  - from **0 (default value)** to 255, with steps of 1
- If the format object is signed:
  - from -128 to 127 with step of 1, **0 (default value)**
- If the format object is percentage:
  - from **0% (default value)** to 100% with step of 5%
- If the format object is two's complement percentage:
  - from -128 to 127 with step of 1, **0 (default value)**
- If the format object is HVAC mode:
  - **auto** **(default value)**
  - comfort
  - precomfort
  - economy
  - off (building protection)
- If the format object is scene:
  - **execute scene 0** **(default value)**
  - execute scene 1
    …
  - execute scene 63
  - learn scene 0
  - learn scene 1
    …
  - learn scene 63
- If the format object is angular width (degrees):
  - from **0° (default value)** to 360°

- If the object size is **2 bytes**:

  - **unsigned** **(default value)**
  - signed
  - time in seconds
  - time in minutes
  - time in hours
  - floating temperature °C
  - floating temperature °K
  - floating temperature °F
  - floating brightness
  - floating humidity
  - floating solar intensity
  - floating voltage
  - floating current
  - floating power
  - floating wind speed

Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the comparison is "true" or "false", depending on the result to which the parameter refers; the "floating" values could differ from what is actually sent due to conversion. The values that can be set are:

- If the object format is unsigned, time in seconds, time in minutes or time in hours:
  - from **0 (default value)** to 65535, with steps of 1
- If the format object is signed:
  - from **-**32768 to 32767 with step of 1, **0 (default value)**
- the format object is floating °C:
  - from -273 to 670760 with step of 1 for the whole part, **0 (default value)**
  - from **0 (default value)** to 99 with step of 1 for the decimal part
- If the format of the object is floating °K, floating solar intensity, floating voltage, floating current and floating power:

  - from -670760 to 670760 with step of 1 for the whole part, **0 (default value)**
  - from **0 (default value)** to 99 with step of 1 for the decimal part
- the format object is floating °F:
  - from -459 to 670760 with step of 1 for the whole part, **0 (default value)**
  - from **0 (default value)** to 99 with step of 1 for the decimal part
- If the object format is floating brightness, floating humidity or floating wind speed:
  - from **0 (default value)** to 670760 with step of 1 for the whole part
  - from **0 (default value)** to 99 with step of 1 for the decimal part

- If the object size is **3 bytes**:

  - **RGB** **(default value)**

  this setting displays the parameters **"RED component value"**, **"GREEN component value"** and **"BLUE component value"** which are used to set the value of each component to be sent when the result of the comparison is "true" or "false", depending on the result to which the parameter refers. The values that can be set are:

  - from **0 (default value)** to 255, with steps of 1

- If the object size is **4 bytes**:

  - **unsigned** **(default value)**
  - signed
  - signed active power
  - signed reactive power
  - signed apparent power
  - signed active power (kWh)
  - signed reactive power (kVARh)
  - signed apparent power (kVAh)
  - floating power
  - floating power factor
  - floating speed
  - floating volume

  Based on the value set for this item, the values that can be set for the parameter **"Value"** will change as a result, which is used to set the value to be sent when the result of the comparison is "true" or "false", depending on the result to which the parameter refers; the "floating" values could differ from what is actually sent due to conversion. The values that can be set are:

  - If the format object is unsigned:
    - from **0 (default value)** to 4294967295, with steps of 1
  - If the object format is signed, with signed active power, with signed reactive power, with signed apparent power, signed active power(kWh), with signed reactive power(kVARh) or with signed apparent power (kVAh):
    - from -2147483648 to 2147483647 with steps of 1, **0 (default value)**

  - If the object format is floating power, floating power factor, floating speed or floating volume:
    - from -2147483648 to 2147483647 with step of 1 for the whole part, **0 (default value)**
    - from **0 (default value)** to 99 with step of 1 for the decimal part

In addition to the possibility of sending commands, it is possible to associate the activation of the buzzer present on the device to the result of the comparison via the parameter **"Acoustic signal (buzzer)"**, which permits both enabling as well as the possible activation conditions; the values that can be set are:

- **disabled** **(default value)**
- enabled if the event is TRUE
- enabled if the event is FALSE
- enabled for both events

11 Communication objects

Common functions for blocks 1 .. 42					
ETS Function	**Index**	**Name**	**Object function**	**Size**	**Datapoint Type**
1 dimmer	0	Block 1 - Dimmer switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1 - Dimmer block	Active/Deactivate	1 bit	1.002 DPT_Bool
		Block 1 - Dimmer priority command	On/Off forced positioning	2 bits	2.001 DPT_Switch_Control
	2	Block 1 - Dimmer brightness regulation	Increase/Decrease	4 bits	3.007 DPT_Control_Dimming
	3	Block 1 - Dimmer brightness value command	% Value	1 byte	5.001 DPT_Scaling
	4	Block 1 - Dimmer status feedback	On/Off status	1 bit	1.001 DPT_Switch
	5	Block 1 - Dimmer brightness value feedback	% Value	1 byte	5.001 DPT_Scaling
2 dimmers	0	Block 1A - Dimmer switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1A - Dimmer brightness value command	% Value	1 byte	5.001 DPT_Scaling
		Block 1A - Dimmer brightness regulation	Increase/Decrease	4 bits	3.007 DPT_Control_Dimming
	2	Block 1A - Dimmer brightness value feedback	% Value	1 byte	5.001 DPT_Scaling
		Block 1A - Dimmer status feedback	On/Off status	1 bit	1.001 DPT_Switch
	3	Block 1B - Dimmer switching	On/Off	1 bit	1.001 DPT_Switch
	4	Block 1B - Dimmer brightness value command	% Value	1 byte	5.001 DPT_Scaling
		Block 1B - Dimmer brightness regulation	Increase/Decrease	4 bits	3.007 DPT_Control_Dimming
	5	Block 1B - Dimmer brightness value feedback	% Value	1 byte	5.001 DPT_Scaling
		Block 1B - Dimmer status feedback	On/Off status	1 bit	1.001 DPT_Switch
3 dimmers	0	Block 1A - Dimmer brightness regulation	Increase/Decrease	4 bits	3.007 DPT_Control_Dimming
		Block 1A - Dimmer brightness value command	% Value	1 byte	5.001 DPT_Scaling
	1	Block 1A - Dimmer status feedback	On/Off status	1 bit	1.001 DPT_Switch
		Block 1A - Dimmer brightness value feedback	% Value	1 byte	5.001 DPT_Scaling
	2	Block 1B - Dimmer brightness regulation	Increase/Decrease	4 bits	3.007 DPT_Control_Dimming
		Block 1B - Dimmer brightness value command	% Value	1 byte	5.001 DPT_Scaling
	3	Block 1B - Dimmer status feedback	On/Off status	1 bit	1.001 DPT_Switch
		Block 1B - Dimmer brightness value feedback	% Value	1 byte	5.001 DPT_Scaling
	4	Block 1C - Dimmer brightness regulation	Increase/Decrease	4 bits	3.007 DPT_Control_Dimming
		Block 1C - Dimmer brightness value command	% Value	1 byte	5.001 DPT_Scaling
	5	Block 1C - Dimmer status feedback	On/Off status	1 bit	1.001 DPT_Switch
		Block 1C - Dimmer brightness value feedback	% Value	1 byte	5.001 DPT_Scaling

DALI dimmer	0	Block 1 - DALI dimmer/group switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1 - DALI dimmer/group error feedback	Error/No error	1 bit	1.002 DPT_Bool
	2	Block 1 - DALI dimmer/group brightness regulation	Increase/Decrease	4 bits	3.007 DPT_Control_Dimming
	3	Block 1 - DALI dimmer/group brightness command	% Value	1 byte	5.001 DPT_Scaling
	4	Block 1 - DALI dimmer/group status feedback	On/Off status	1 bit	1.001 DPT_Switch
	5	Block 1 - DALI dimmer/group brightness feedback	% Value	1 byte	5.001 DPT_Scaling
1 RGB dimmer	0	Block 1A - RGB red brightness value command	Value (0 - 255)	1 byte	5.010 DPT_Value_1_Ucount
	1	Block 1A - RGB red brightness value feedback	Value (0 - 255)	1 byte	5.010 DPT_Value_1_Ucount
	2	Block 1A - RGB green brightness value command	Value (0 - 255)	1 byte	5.010 DPT_Value_1_Ucount
	3	Block 1B - RGB green brightness value feedback	Value (0 - 255)	1 byte	5.010 DPT_Value_1_Ucount
	4	Block 1A - RGB blue brightness value command	Value (0 - 255)	1 byte	5.010 DPT_Value_1_Ucount
	5	Block 1C - RGB blue brightness value feedback	Value (0 - 255)	1 byte	5.010 DPT_Value_1_Ucount
3 RGB dimmers	0	Block 1A - RGB brightness value command	RGB colors 0 - 255 values	3 bytes	232.600 DPT_Colour_RGB
	1	Block 1A - RGB brightness value feedback	RGB colors 0 - 255 values	3 bytes	232.600 DPT_Colour_RGB
	2	Block 1B - RGB brightness value command	RGB colors 0 - 255 values	3 bytes	232.600 DPT_Colour_RGB
	3	Block 1B - RGB brightness value feedback	RGB colors 0 - 255 values	3 bytes	232.600 DPT_Colour_RGB
	4	Block 1C - RGB brightness value command	RGB colors 0 - 255 values	3 bytes	232.600 DPT_Colour_RGB
	5	Block 1C - RGB brightness value feedback	RGB colors 0 - 255 values	3 bytes	232.600 DPT_Colour_RGB
1 light	0	Block 1 - Light switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1 - Light priority command	On/Off forced positioning	2 bits	2.001 DPT_Switch_Control
	2	Block 1 - Light block	Active/Deactivate	1 bit	1.002 DPT_Bool
	3	Block 1 - Light status feedback	On/Off status	1 bit	1.001 DPT_Switch
	4	Not used			
	5	Not used			
2 lights	0	Block 1A - Light switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1A - Light block	Active/Deactivate	1 bit	1.002 DPT_Bool
		Block 1A - Light priority command	On/Off forced positioning	2 bits	2.001 DPT_Switch_Control
	2	Block 1A - Light status feedback	On/Off status	1 bit	1.001 DPT_Switch
	3	Block 1B - Light switching	On/Off	1 bit	1.001 DPT_Switch
	4	Block 1B - Light block	Active/Deactivate	1 bit	1.002 DPT_Bool
		Block 1B - Light priority command	On/Off forced positioning	2 bits	2.001 DPT_Switch_Control
	5	Block 1B - Light status feedback	On/Off status	1 bit	1.001 DPT_Switch
3 lights	0	Block 1A - Light switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1A - Light status feedback	On/Off status	1 bit	1.001 DPT_Switch
	2	Block 1B - Light switching	On/Off	1 bit	1.001 DPT_Switch
	3	Block 1B - Light status feedback	On/Off status	1 bit	1.001 DPT_Switch
	4	Block 1C - Light switching	On/Off	1 bit	1.001 DPT_Switch
	5	Block 1C - Light status feedback	On/Off status	1 bit	1.001 DPT_Switch

1on/off actuator	0	Block 1 - On/Off actuator switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1 - On/Off actuator priority command	On/Off forced positioning	2 bits	2.001 DPT_Switch_Control
	2	Block 1 - On/Off actuator block	Active/Deactivate	1 bit	1.002 DPT_Bool
	3	Block 1 - On/Off actuator status feedback	On/Off status	1 bit	1.001 DPT_Switch
	4	Not used			
	5	Not used			
2 on/off actuators	0	Block 1A - On/Off actuator switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1A - On/Off actuator block	Active/Deactivate	1 bit	1.002 DPT_Bool
		Block 1A - On/Off actuator priority command	On/Off forced positioning	2 bits	2.001 DPT_Switch_Control
	2	Block 1A - On/Off actuator status feedback	On/Off status	1 bit	1.001 DPT_Switch
	3	Block 1B - On/Off actuator switching	On/Off	1 bit	1.001 DPT_Switch
	4	Block 1B - On/Off actuator block	Active/Deactivate	1 bit	1.002 DPT_Bool
		Block 1B - On/Off actuator priority command	On/Off forced positioning	2 bits	2.001 DPT_Switch_Control
	5	Block 1B - On/Off actuator status feedback	On/Off status	1 bit	1.001 DPT_Switch
3 on/off actuators	0	Block 1A - On/Off actuator switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1A - On/Off actuator status feedback	On/Off status	1 bit	1.001 DPT_Switch
	2	Block 1B - On/Off actuator switching	On/Off	1 bit	1.001 DPT_Switch
	3	Block 1B - On/Off actuator status feedback	On/Off status	1 bit	1.001 DPT_Switch
	4	Block 1C - On/Off actuator switching	On/Off	1 bit	1.001 DPT_Switch
	5	Block 1C - On/Off actuator status feedback	On/Off status	1 bit	1.001 DPT_Switch
1 shutter	0	Block 1 - Shutter movement	Up/Down	1 bit	1.008 DPT_UpDown
	1	Block 1 - Shutter stop/Louvres control	Stop/Step	1 bit	1.007 DPT_Step
	2	Block 1 - Shutter position command	% Value	1 byte	5.001 DPT_Scaling
	3	Block 1 - Shutter priority command	Up/Down forced positioning	2 bits	2.008 DPT_Direction1_Control
	4	Block 1 - Shutter block	Active/Deactivate	1 bit	1.002 DPT_Bool
	5	Block 1 - Shutter position feedback	% Value	1 byte	5.001 DPT_Scaling
2 shutters	0	Block 1A - Shutter movement	Up/Down	1 bit	1.008 DPT_UpDown
	1	Block 1 - Shutter stop/Louvres control	Stop/Step	1 bit	1.007 DPT_Step
	2	Block 1A - Shutter position feedback	% Value	1 byte	5.001 DPT_Scaling
	3	Block 1B - Shutter movement	Up/Down	1 bit	1.008 DPT_UpDown
	4	Block 1 - Shutter stop/Louvres control	Stop/Step	1 bit	1.007 DPT_Step
	5	Block 1B - Shutter position feedback	% Value	1 byte	5.001 DPT_Scaling

3 shutters	0	Block 1A - Shutter movement	Up/Down	1 bit	1.008 DPT_UpDown
		Block 1A - Shutter position command	% Value	1 byte	5.001 DPT_Scaling
	1	Block 1A - Shutter stop/Louvres control	Stop/Step	1 bit	1.007 DPT_Step
		Block 1A - Shutter position feedback	% Value	1 byte	5.001 DPT_Scaling
	2	Block 1B - Shutter movement	Up/Down	1 bit	1.008 DPT_UpDown
		Block 1B - Shutter position command	% Value	1 byte	5.001 DPT_Scaling
	3	Block 1B - Shutter stop/Louvres control	Stop/Step	1 bit	1.007 DPT_Step
		Block 1B - Shutter position feedback	% Value	1 byte	5.001 DPT_Scaling
	4	Block 1C - Shutter movement	Up/Down	1 bit	1.008 DPT_UpDown
		Block 1C - Shutter position command	% Value	1 byte	5.001 DPT_Scaling
	5	Block 1C - Shutter stop/Louvres control	Stop/Step	1 bit	1.007 DPT_Step
		Block 1C - Shutter position feedback	% Value	1 byte	5.001 DPT_Scaling
HVAC master	0	Block 1 - Functioning type sending	Heating/Air cooling	1 bit	1.100 DPT_Heat/Cool
	1	Block 1 - HVAC mode sending	Auto/Eco/Precom/Comf/Off	1 byte	20.102 DPT_HVACMode
		Block 1 - Setpoint sending	Value (°C)	2 bytes	9.001 DPT_Value_Temp
		Block 1 - Setpoint sending	Value (°K)	2 bytes	9.002 DPT_Value_Tempd
		Block 1 - Setpoint sending	Value (°F)	2 bytes	9.027 DPT_Value_Temp_F
	2	Block 1 - Measured temperature feedback	Value (°C)	2 bytes	9.001 DPT_Value_Temp
		Block 1 - Measured temperature feedback	Value (°K)	2 bytes	9.002 DPT_Value_Tempd
		Block 1 - Measured temperature feedback	Value (°F)	2 bytes	9.027 DPT_Value_Temp_F
	3	Block 1 - Setpoint feedback	Value (°C)	2 bytes	9.001 DPT_Value_Temp
		Block 1 - Setpoint feedback	Value (°K)	2 bytes	9.002 DPT_Value_Tempd
		Block 1 - Setpoint feedback	Value (°F)	2 bytes	9.027 DPT_Value_Temp_F
	4	Block 1 - Functioning type feedback	Heating/Air cooling	1 bit	1.100 DPT_Heat/Cool
	5	Block 1 - HVAC mode feedback	Auto/Eco/Precom/Comf/Off	1 byte	20.102 DPT_HVACMode
KNX scenes	0	Block 1A - KNX scene	Execute/Learn	1 byte	18.001 DPT_SceneControl
		Block 1A - KNX scene	Execute	1 byte	17.001 DPT_SceneNumber
	1	Block 1A - Execute KNX scene command trigger	Sending command trigger	1 bit	1.017 DPT_Trigger
	2	Block 1B - KNX scene	Execute/Learn	1 byte	18.001 DPT_SceneControl
		Block 1B - KNX scene	Execute	1 byte	17.001 DPT_SceneNumber
	3	Block 1B - Execute KNX scene command trigger	Sending command trigger	1 bit	1.017 DPT_Trigger
	4	Block 1C - KNX scene	Execute/Learn	1 byte	18.001 DPT_SceneControl
		Block 1C - KNX scene	Execute	1 byte	17.001 DPT_SceneNumber
	5	Block 1C - Execute KNX scene command trigger	Sending command trigger	1 bit	1.017 DPT_Trigger

irrigation	0	Block 1A - Irrigator switching	On/Off	1 bit	1.001 DPT_Switch
	1	Block 1A - Irrigator status feedback	On/Off status	1 bit	1.001 DPT_Switch
	2	Block 1A - Irrigator block	1=Active/0=Deactivate	1 bit	1.003 DPT_Enable
	3	Block 1B - Irrigator switching	On/Off	1 bit	1.001 DPT_Switch
	4	Block 1B - Irrigator status feedback	On/Off status	1 bit	1.001 DPT_Switch
	5	Block 1B - Irrigator block	1=Active/0=Deactivate	1 bit	1.003 DPT_Enable
6 independent inputs	0 = Block 1A 1 = Block 1B 2 = Block 1C 3 = Block 1D 4 = Block 1E 5 = Block 1F	Block 1A - 1 bit input	1=On/0=Off	1 bit	1.001 DPT_Switch
			1=True/0=False	1 bit	1.002 DPT_Bool
			1=Heating/0=Air Cooling	1 bit	1.100 DPT_Heat/Cool
			1=Enable/0=Disable	1 bit	1.003 DPT_Enable
			1=Alarm/0=No alarm	1 bit	1.005 DPT_Alarm
			1=Occupied/0=Free	1 bit	1.018 DPT_Occupancy
			1=Close/0=Open	1 bit	1.009 DPT_OpenClose
			Input with surveillance	1 bit	1.001 DPT_Switch
			Window 1=Open/0=Closed	1 bit	1.019 DPT_Window/Door
		Block 1A - 1 byte input	Unsigned value	1 byte	5.010 DPT_Value_1_Ucount
			Signed value	1 byte	6.010 DPT_Value_1_Count
			% Value	1 byte	5.001 DPT_Scaling
			Two's complement % value	1 byte	6.001 DPT_Percent_V8
			Auto/Eco/Precom/Comf/Off	1 byte	20.102 DPT_HVACMode
		Block 1A - 2 bytes input	Unsigned value	2 bytes	7.001 DPT_Value_2_Ucount
			Signed value	2 bytes	8.001 DPT_Value_2_Count
			Time (seconds)	2 bytes	7.005 DPT_TimePeriodSec
			Time (minutes)	2 bytes	7.006 DPT_TimePeriodMin
			Time (hours)	2 bytes	7.006 DPT_TimePeriodMin
			Value (°C)	2 bytes	9.001 DPT_Value_Temp
			Value (°K)	2 bytes	9.002 DPT_Value_Tempd
			Value (°F)	2 bytes	9.027 DPT_Value_Temp_F
			Value (lux)	2 bytes	9.004 DPT_Value_Lux
			% Humidity value	2 bytes	9.007 DPT_Value_Humidity
			Power density (W/m^2)	2 bytes	9.022 DPT_PowerDensity
			Value (milliVolt)	2 bytes	9.020 DPT_Value_Volt
			Value (milliAmpere)	2 bytes	9.021 DPT_Value_Curr
			Value (kiloWatt)	2 bytes	9.024 DPT_Power
		Block 1A - 3 bytes input	Day/Time	3 bytes	10.001 DPT_TimeOfDay
			Date	3 bytes	11.001 DPT_Date
			RGB color	3 bytes	232.600 DPT_Colour_RGB
		Block 1A - 4 bytes input	Unsigned value	4 bytes	12.001 DPT_Value_4_Ucount
			Signed value	4 bytes	13.001 DPT_Value_4_Count
			Value (Wh)	4 bytes	13.010 DPT_ActiveEnergy

			Value (VARh)	4 bytes	13.012 DPT_ReactiveEnergy
			Value (VAh)	4 bytes	13.011 DPT_ApparantEnergy
			Value (kWh)	4 bytes	13.013 DPT_ActiveEnergy_kWh
			Value (kVARh)	4 bytes	13.015 DPT_ReactiveEnergy_kVARh
			Value (kVAh)	4 bytes	13.014 DPT_ApparantEnergy_kVAh
			Value (Watt)	4 bytes	14.056 DPT_Value_Power
			Value (cosΦ)	4 bytes	14.057 DPT_Value_Power_Factor
			Value (meters per second)	4 bytes	14.065 DPT_Value_Speed
			Value (cubic meters)	4 bytes	14.076 DPT_Value_Volume
6 independent outputs	0 = Block 1A 1 = Block 1B 2 = Block 1C 3 = Block 1D 4 = Block 1E 5 = Block 1F	Block 1A - 1 bit output	1=On/0=Off	1 bit	1.001 DPT_Switch
			1=True/0=False	1 bit	1.002 DPT_Bool
			1=Down/0=Up	1 bit	1.008 DPT_UpDown
			1=Increase/0=Decrease	1 bit	1.007 DPT_Step
			1=Heating/0=Air Cooling	1 bit	1.100 DPT_Heat/Cool
			1=Enable/0=Disable	1 bit	1.003 DPT_Enable
			1=Alarm/0=No alarm	1 bit	1.005 DPT_Alarm
			1=Start/0=Stop	1 bit	1.010 DPT_Start
			1=Close/0=Open	1 bit	1.9 DPT_OpenClose
		Block 1A - 2 bits output	On/Off priority command	2 bits	2.001 DPT_Switch_Control
			Up/Down priority command	2 bits	2.008 DPT_Direction1_Control
		Block 1A - 4 bits output	Step dimmer	4 bits	3.007 DPT_Control_Dimming
			Step shutter/venetian	4 bits	3.008 DPT_Control_Blinds
		Block 1A - 1 byte output	Unsigned value	1 byte	5.010 DPT_Value_1_Ucount
			Signed value	1 byte	6.010 DPT_Value_1_Count
			% Value	1 byte	5.001 DPT_Scaling
			Two's complement % value	1 byte	6.001 DPT_Percent_V8
			Auto/Eco/Precom/Comf/Off	1 byte	20.102 DPT_HVACMode
		Block 1A - 2 bytes output	Unsigned value	2 bytes	7.001 DPT_Value_2_Ucount
			Signed value	2 bytes	8.001 DPT_Value_2_Count
			Time (seconds)	2 bytes	7.005 DPT_TimePeriodSec
			Time (minutes)	2 bytes	7.006 DPT_TimePeriodMin
			Time (hours)	2 bytes	7.006 DPT_TimePeriodMin
			Value (°C)	2 bytes	9.001 DPT_Value_Temp
			Value (°K)	2 bytes	9.002 DPT_Value_Tempd
			Value (°F)	2 bytes	9.027 DPT_Value_Temp_F
			Value (lux)	2 bytes	9.004 DPT_Value_Lux
			% Humidity value	2 bytes	9.007 DPT_Value_Humidity
			Power density (W/m^2)	2 bytes	9.027 DPT_Value_Temp_F
			Value (milliVolt)	2 bytes	9.020 DPT_Value_Volt
			Value (milliAmpere)	2 bytes	9.021 DPT_Value_Curr

			Value (kiloWatt)	2 bytes	9.024 DPT_Power
		Block 1A - 3 bytes output	Day/Time	3 bytes	10.001 DPT_TimeOfDay
			Date	3 bytes	11.001 DPT_Date
			RGB color	3 bytes	232.600 DPT_Colour_RGB
		Block 1A - 4 bytes output	Unsigned value	4 bytes	12.001 DPT_Value_4_Ucount
			Signed value	4 bytes	13.001 DPT_Value_4_Count
			Value (Wh)	4 bytes	13.010 DPT_ActiveEnergy
			Value (VARh)	4 bytes	13.012 DPT_ReactiveEnergy
			Value (VAh)	4 bytes	13.011 DPT_ApparantEnergy
			Value (kWh)	4 bytes	13.013 DPT_ActiveEnergy_kWh
			Value (kVARh)	4 bytes	13.015 DPT_ReactiveEnergy_kVARh
			Value (kVAh)	4 bytes	13.014 DPT_ApparantEnergy_kVAh
			Value (Watt)	4 bytes	14.056 DPT_Value_Power
			Value (cosΦ)	4 bytes	14.057 DPT_Value_Power_Factor
			Value (meters per second)	4 bytes	14.065 DPT_Value_Speed
			Value (cubic meters)	4 bytes	14.076 DPT_Value_Volume
6 videoentryphone events	0 = Block 1A 1 = Block 1B 2 = Block 1C 3 = Block 1D 4 = Block 1E 5 = Block 1F	Block 1A - Command to videoentyphone system	KNX -> Videoentyphone	1 bit	1.001 DPT_Switch
		Block 1A - Videoentyphone event feedback	Videoentyphone -> KNX	1 bit	1.001 DPT_Switch
				1 byte	5.010 DPT_Value_1_Ucount
				1 byte	6.010 DPT_Value_1_Count
				1 byte	5.001 DPT_Scaling
				1 byte	6.001 DPT_Percent_V8
				1 byte	20.102 DPT_HVACMode
				1 byte	18.001 DPT_SceneControl

In the table are listed the communication objects related to Block 1; for Blocks 2 .. 42, the objects are the same except for indexes that are a function of the number of the block.

Alternative specific functions

ETS Function	Index	Name	Object function	Size	Datapoint Type
Date and Time (Block 15)	84	Date and Time - Date input	Receive day/month/year	3 bytes	11.001 DPT_Date
	85	Date and Time - Date output	Send day/month/year	3 bytes	11.001 DPT_Date
	86	Date and Time - Time of day input	Receive time/day	3 bytes	10.001 DPT_TimeOfDay
	87	Date and Time - Time of day output	Send time/day	3 bytes	10.001 DPT_TimeOfDay
	88	Date and Time - Daylight saving time autom. update	Active/Deactivate	1 bit	1.002 DPT_Bool
	89	Date and Time - Daylight saving time output	1=daylight sav time/0=standard	1 bit	1.002 DPT_Bool

Chronotermostat zone1 (Blocks 16-17)	90	Chrono 1 - HVAC mode input	Auto/Eco/Precom/Comf/Off	1 byte	20.102 DPT_HVACMode
	91	Chrono 1 - Functioning type input	Heating/Air cooling	1 bit	1.100 DPT_Heat/Cool
	92	Chrono 1 - Measured temperature input	Value (°C)	2 bytes	9.001 DPT_Value_Temp
	92	Chrono 1 - Measured temperature input	Value (°K)	2 bytes	9.002 DPT_Value_Tempd
	92	Chrono 1 - Measured temperature input	Value (°F)	2 bytes	9.027 DPT_Value_Temp_F
	93	Chrono 1 - Measured humidity input	% Value	2 bytes	9.007 DPT_Value_Humidity
	94	Chrono 1 - Functioning type feedback	Heating/Air cooling	1 bit	1.100 DPT_Heat/Cool
	95	Chrono 1 - HVAC mode feedback	Auto/Eco/Precom/Comf/Off	1 byte	20.102 DPT_HVACMode
	96	Chrono 1 - Current setpoint feedback	Value (°C)	2 bytes	9.001 DPT_Value_Temp
	96	Chrono 1 - Current setpoint feedback	Value (°K)	2 bytes	9.002 DPT_Value_Tempd
	96	Chrono 1 - Current setpoint feedback	Value (°F)	2 bytes	9.027 DPT_Value_Temp_F
	97	Chrono 1 - Heating/cooling valve status feedback	On/Off status	1 bit	1.001 DPT_Switch
	97	Chrono 1 - Heating valve status feedback	On/Off status	1 bit	1.001 DPT_Switch
	98	Chrono 1 - Heating/Cooling valve switch	On/Off	1 bit	1.001 DPT_Switch
	98	Chrono 1 - Heating valve switch	On/Off	1 bit	1.001 DPT_Switch
	99	Chrono 1 - Cooling valve status feedback	On/Off status	1 bit	1.001 DPT_Switch
	100	Chrono 1 - Cooling valve switch	On/Off	1 bit	1.001 DPT_Switch
	101	Not used			
Zone 1 Fan-coil heating (Block 18)	102	Chrono 1 - Heating fan V1 status feedback	On/Off status	1 bit	1.001 DPT_Switch
	103	Chrono 1 - V1 fan switching heating	On/Off	1 bit	1.001 DPT_Switch
	104	Chrono 1 - Heating fan V2 status feedback	On/Off status	1 bit	1.001 DPT_Switch
	105	Chrono 1 - V2 fan switching heating	On/Off	1 bit	1.001 DPT_Switch
	106	Chrono 1 - Heating fan V3 status feedback	On/Off status	1 bit	1.001 DPT_Switch
	107	Chrono 1 - V3 fan switching heating	On/Off	1 bit	1.001 DPT_Switch
Zone 1 Fan-coil air cooling (Block 19)	108	Chrono 1 - Cooling fan V1 status feedback	On/Off status	1 bit	1.001 DPT_Switch
	109	Chrono 1 - V1 fan switching cooling	On/Off	1 bit	1.001 DPT_Switch
	110	Chrono 1 - Cooling fan V2 status feedback	On/Off status	1 bit	1.001 DPT_Switch
	111	Chrono 1 - V2 fan switching cooling	On/Off	1 bit	1.001 DPT_Switch
	112	Chrono 1 - Cooling fan V3 status feedback	On/Off status	1 bit	1.001 DPT_Switch
	113	Chrono 1 - V3 fan switching cooling	On/Off	1 bit	1.001 DPT_Switch
Chronotermostat zone 2 (Blocks 20-21)	114	see Blocks 16-17			
	115				
	..				
	124				
	125				

Zone 2 Fan-coil heating (Block 22)	126	see Block 18
	127	
	..	
	130	
	131	
Zone 2 Fan-coil air cooling (Block 23)	132	see Block 19
	133	
	..	
	136	
	137	
Chronotermostat zone 3 (Block 24-25)	138	see Blocks 16-17
	139	
	..	
	148	
	149	
Zone 3 Fan-coil heating (Block 26)	150	see Block 18
	151	
	..	
	154	
	155	
Zone 3 Fan-coil air cooling (Block 27)	156	see Block 19
	157	
	..	
	160	
	161	
Chronotermostat zone 4 (Blocks 28-29)	162	see Blocks 16-17
	163	
	..	
	172	
	173	
Zone 4 Fan-coil heating (Block 30)	174	see Block 18
	175	
	..	
	178	
	179	
Zone 4 Fan-coil air cooling (Block 31)	180	see Block 19
	181	
	..	
	184	
	185	

Burglar system - Area 14 control (Blocks 32-33)	186	Burglar - Area 4 sector 1 activation status	Activated/Deactivated	1 bit	1.001 DPT_Switch
	187	Burglar - Area 4 sector 1 command	Activation/Deactivation	1 bit	1.001 DPT_Switch
	188	Burglar - Area 4 sector 2 activation status	Activated/Deactivated	1 bit	1.001 DPT_Switch
	189	Burglar - Area 4 sector 2 command	Activation/Deactivation	1 bit	1.001 DPT_Switch
	190	Burglar - Area 4 sector 3 activation status	Activated/Deactivated	1 bit	1.001 DPT_Switch
	191	Burglar - Area 4 sector 3 command	Activation/Deactivation	1 bit	1.001 DPT_Switch
	192	Burglar - Area 4 sector 4 activation status	Activated/Deactivated	1 bit	1.001 DPT_Switch
	193	Burglar - Area 4 sector 4 command	Activation/Deactivation	1 bit	1.001 DPT_Switch
	194	Burglar - Area 4 sector 1 activation enabling	Enabled/Disabled	1 bit	1.003 DPT_Enable
	195	Burglar - Area 4 sector 2 activation enabling	Enabled/Disabled	1 bit	1.003 DPT_Enable
	196	Burglar - Area 4 sector 3 activation enabling	Enabled/Disabled	1 bit	1.003 DPT_Enable
	197	Burglar - Area 4 sector 4 activation enabling	Enabled/Disabled	1 bit	1.003 DPT_Enable
Burglar system - Area 3 control (Blocks 34-35)	198	see Blocks 32 - 33			
	199				
	..				
	202				
	203				
Burglar system - Area 2 control (Blocks 36-37)	204	see Blocks 32 - 33			
	205				
	..				
	220				
	221				
Burglar system - Area 1 control (Blocks 38-39)	222	see Blocks 32 - 33			
	223				
	..				
	232				
	233				
Burglar system - Total control and alarms feedback (Blocks 40-41)	234	Burglar - Global activation status	Activated/Deactivated	1 bit	1.001 DPT_Switch
	235	Burglar - Global activation command	Activation/Deactivation	1 bit	1.001 DPT_Switch
	236	Burglar - Global activation enabling	Enabled/Disabled	1 bit	1.003 DPT_Enable
	237	Burglar - Global alarm feedback	On/Off alarm	1 bit	1.005 DPT_Alarm
	238	Burglar - Area 1 alarm feedback	On/Off alarm	1 bit	1.005 DPT_Alarm
	239	Burglar - Area 2 alarm feedback	On/Off alarm	1 bit	1.005 DPT_Alarm
	240	Burglar - Area 3 alarm feedback	On/Off alarm	1 bit	1.005 DPT_Alarm
	241	Burglar - Area 4 alarm feedback	On/Off alarm	1 bit	1.005 DPT_Alarm
	242	Not used			
	243	Not used			
	244	Not used			
	245	Not used			

Energy management - Load control (Block 42)	246	Energy management - Active energy consumed [kWh]	Value (kWh)	4 bytes	13.013 DPT_ActiveEnergy_kWh
		Energy management - Active energy consumed [Wh]	Value (Wh)	4 bytes	13.010 DPT_ActiveEnergy
	247	Energy management - Active energy produced [kWh]	Value (kWh)	4 bytes	13.013 DPT_ActiveEnergy_kWh
		Energy management - Active energy produced [Wh]	Value (Wh)	4 bytes	13.010 DPT_ActiveEnergy
	248	Energy management - Active power consumed [kW]	Value (kW)	2 bytes	9.024 DPT_Power
		Energy management - Active power consumed [W]	Value (W)	4 bytes	14.056 DPT_Value_Power
		Energy management - Active signed power [kW]	+ consumed | - produced	2 bytes	9.024 DPT_Power
		Energy management - Active signed power [W]	+ consumed | - produced	4 bytes	14.056 DPT_Value_Power
		Energy management - Active signed power [kW]	+ produced | - consumed	2 bytes	9.024 DPT_Power
		Energy management - Active signed power [W]	+ produced | - consumed	4 bytes	14.056 DPT_Value_Power
		Energy management - Active power [kW]	Value (kW)	2 bytes	9.024 DPT_Power
		Energy management - Active power [W]	Value (W)	4 bytes	14.056 DPT_Value_Power
	249	Energy management - Active power/Load type	Consumed/Produced Ind/Cap	1 byte	Non standard DPT
		Energy management - Active power produced [kW]	Value (kW)	2 bytes	9.024 DPT_Power
		Energy management - Active power produced [W]	Value (W)	4 bytes	14.056 DPT_Value_Power
	250	Energy management - Water counter	Value (m^3)	4 bytes	14.076 DPT_Value_Volume
	251	Energy management - Gas counter	Value (m^3)	4 bytes	14.076 DPT_Value_Volume

cod. 7.01.6.745.7

Ai sensi dell'articolo 9 comma 2 della Direttiva Europea 2004/108/CE si informa che responsabile dell'immissione del prodotto sul mercato Comunitario è:
According to article 9 paragraph 2 of the European Directive 2004/108/EC, the responsible for placing the apparatus on the Community market is:
GEWISS S.p.A Via A. Volta, 1 - 24069 Cenate Sotto (BG) Italy Tel: +39 035 946 111 Fax: +39 035 945 270 E-mail: qualitymarks@gewiss.com

SAT
+39 035 946 111
8.30 - 12.30 / 14.00 - 18.00
lunedì ÷ venerdì - monday ÷ friday
24h
+39 035 946 260
sat@gewiss.com
www.gewiss.com

LAST REVISION 02/2014