Technical Documentation Intensity

Status: Draft
Firmware Version : v1.4
Hardware Version : Intensity Snr. 140505 / 140203
Order Number: INTXXX

Technical Details:

Power Supply	Min. 21 VDC, Max. 31 VDC, provided by the KNX bus line
Maximum Power Consumption	INT/INF: 650 mW
Bus Power Consumption	Class 30mA
Fan-In Model	TP1-256
Connection to the bus via:	2 x 1 mm pins for bus connecting terminal (TP1),0.5mm2 section
Operating Temperatures	-5 ºC to +45 ºC
Maximum Humidity	93% relative humidity, no moisture condensation
Type Of Protection (EN 60529)	IP20 (with front plate mounted)
Protective Separation	Device Group 3
Dimensions (w x h x d) (mm)	+/- 85 x 85 x 30 (metal) +/- 90 x 90 x 30 (glass/stone)

Functional Description

The Intensity consists of 1, 2, 4 or 8 touch surfaces that can be used individually. They can be set to perform various functions, allowing you, for example, to switch lighting on and off, dim it, control the blinds, recall or save scenes, etc

The functions of the INTENSITY are:

- Switch or Send 1 or 2 Byte Values on
  - Single touch
  - Short and Long Touch
  - Positive/Negative Edges
- Dimming (using 1 or 2 buttons)
- Blind Control (using 1 or 2 buttons)
- Shutter Control(using 1 or 2 buttons), with predefined operation concepts:
  - Short Touch : start / long:stop
  - Long Touch : start / short:stop
  - Single Touch : (short: start/stop)
- Recalling / Saving Scenes
- Color LED feedback control

This functionality is further made complete by some functional modules

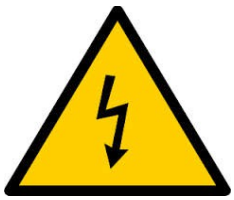
- Scene Module
- Basic Logic Functions
- Timers
- Up/Down Counters

The functions and parameters are explained in more detail hereafter.

The parameters are divided into 3 Parts

1. Button Configuration
   Here you can set what function is behind every touch surface. You can also configure the behavior of the LED that is integrated in every touch surface.
2. Temperature
   Contains the settings of the internal temperature sensor, as well as those for the thermostat.
3. Modules
   Activate additional functionality that comes with each switch. Currently supported:
   - Scene Module, supporting eight scenes with eight actuators (1 Bit/1Byte/2Byte supported)
   - Basic Logic Module (AND/OR functions), consisting of five Logic Channels that each have up to five 1-Bit inputs
   - Timers, up to four
   - Up/Down Counters, up to four

Installation



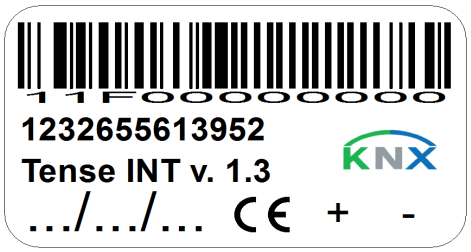
Risk of electrocution

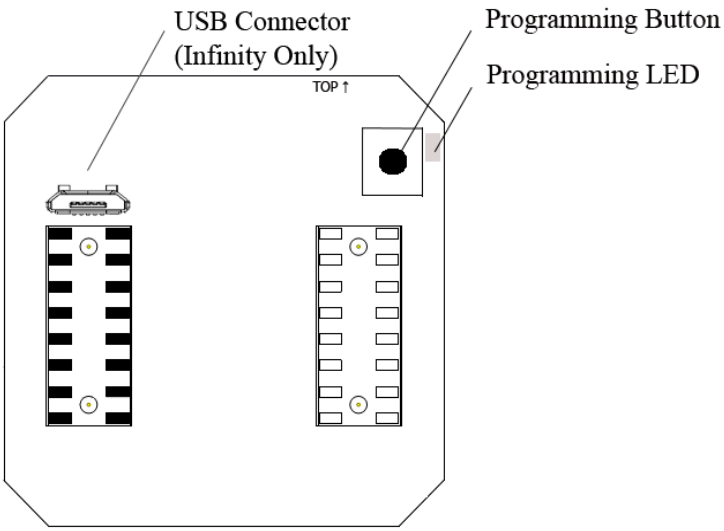
Only skilled electricians can carry out installation and commissioning of the device. Otherwise, there is a risk of fire and electrocution. Observe the regulations valid in the country of use, as well as the KNX guidelines. To be installed indoors.

1. Remove power from the KNX bus.
2. Connect the bus coupler with the KNX bus using the KNX TP1 bus connection terminal. Connect the red bus wire to the red terminal (+) and the black bus wire to the black terminal (–).
3. The bus coupler fits in a standard size 60-installation box .Use two screws to fix the bus coupler. Make sure the mounting is level and that the "TOP ↑" marking on the PCB points upwards.
4. Plug the front onto the bus coupler. Make sure that the "TOP ↑" marking on the PCB (backside of the front) points upwards.
5. Power the KNX bus.

Commissioning

First, download the appropriate product databases from www.tense.be and import it into the ETS.
If you want to avoid a download of the firmware the first time, you can use the firmware version that matches the preprogrammed firmware. You find the version of the preprogrammed software version on the label at the backside of the bus coupler unit.





1. Remove the front.
2. Press the programming button. Make sure the red programming LED lights up.
3. In the ETS, add the device and assign the physical address.
4. Program the physical address. Make sure the red programming LED turns off.
5. Replug the front.
6. In the ETS, select the appropriate parameters and assign the group addresses.
7. Download the application program to the device.

General Settings

In this page you can set some general settings. e.g. The number of touch surfaces that the front on your Intensity has, whether you want to enable MultiTouch (an extra function when multiple touch surfaces are touched at the same time) and the default brightness level of the color LEDs.

You can also enable a blocking object, which will cause the Intensity to ignore any touch input. The other functions are not affected by this setting and bus communication is still possible. LEDs will not provide visual feedback, but otherwise the behavior of the LEDs will act as programmed.

On this page you can also set the default LED brightness, in steps of 10%. Optionally you can choose to use a night object. When the night object is set, then the brightness will be set to the corresponding parameter.

You can also specify the "Read on Init delay time". The "Read On Init" flag is a communication object flag new for System B devices. If you set this flag on a communication object, then the intensity will issue read requests upon power-up, to make sure that its status values are up-to-date.

Set this value to a time, where you are certain that every bus device is up and running and will answer read requests. In that way you assure that the internal states of the Intensity correspond with the actual values.

Note that for a correct working of the *Read On Init* flag, you must also set the *Communication*, *Update, Transmit* and *Write* flag.

Name	***Value Range***	***Comment***
Number of Physical Touch Surfaces on front plate	1 / 2 / 4 / 8	Number of Touch Surfaces present on the front plate of the Intensity
Enable MultiTouch	Yes/No	Enable an extra function by touching more than one surface at once
MultiTouch Subsequent Key Time (ms)	50 – 500	The maximum time in milliseconds in which a second button must be touched in order to activate the "MultiTouch" function
Default LED Brightness Level	Off, 10-100%, Use Night Object	Default Brightness level, or option to use night object to select the dim level of the LEDs
Level at Night = 1	Off, 10-100%	LED Brightness level at Night
Level at Night = 0	Off, 10-100%	LED Brightness level at Day
Enable Blocking Object	Yes/No	Disable Touch Input
Read On Init Delay[s]	4 – 255	Time To wait before reading objects with the ROI flag

List of available Communication Objects and Parameters:

No	*Name*	*I/O*	*DPT*	*Flags*	*Use*
1	Blocking Object	I	DPT1.2	WCTUI	Enables/Disables Touch Input
2	Internal Temperature	O	DPT9.001	RCT	Measurement of internal NTC sensor
3	Night	I	DPT1.2	WCTUI	Changes the brightness of the LEDS

Button Configuration

Switch + Values

The combination of Switch + Value allows you to send a different value on a Single Touch, short/long button touch, or on touching/releasing (= edge) the button. The parameters for the exact period of a long touch can be set.

Possible value types are 1Bit, 1Byte and 2Byte.

In this way you can use this module to

- turn on/off a 1-Bit actuator
- toggle a 1-Bit actuator
- recall a scene
- move a shutter to a certain position
- adjust the dimming to a fixed absolute value
- etc...

or a combination of any of these:

e.g: short touch toggles a light, while a long touch recalls a scene
or short touch activates temperature comfort zone, long touch turns all lights off, …

List of the parameters and communication objects:

Name	***Value Range***	***Comment***
Touch Selection	Single Touch / Short And Long Touch / Edges	Type of touch to react upon
Value Type Selection	1 bit / 1 Byte / 2 Byte	type of value of the communication object
Action	On / Off / Toggle / None	type of action to take when communication object is 1 bit. None will disable any action
Value	0-255, 0-65535	Value to be transmitted. Range depends on the value type
Long Touch Time (x 100ms)	3 – 50	minimum time a button must be touched to generate a long touch event

Com Objects for the first Button:

No	*Name*	*I/O*	*DPT*	*Flags*	*Use*
4	Short Touch – 1 Bit	O	DPT1.1	CT	Switch Object
4	Pos Edge - 1 Bit	O	DPT1.1	CT	Switch Object
4	Single Touch – 1 Bit	O	DPT1.1	CT	Switch Object
5	Long Touch – 1 Bit	O	DPT1.1	CT	Switch Object
5	Neg Edge - 1 Bit	O	DPT1.1	CT	Switch Object
6	Short Touch- 1 Byte	O	DPT1.1	CT	1 Byte Value
6	Pos Edge - 1 Byte	O	DPT1.1	CT	1 Byte Value
6	Single Touch- 1 Byte	O	DPT1.1	CT	1 Byte Value
7	Long Touch – 1 Byte	O	DPT1.1	CT	1 Byte Value
7	Neg Edge - 1 Byte	O	DPT1.1	CT	1 Byte Value
9	Short Touch- 2 Bytes	O	DPT1.1	CT	2 Bytes Value
9	Pos Edge - 2 Bytes	O	DPT1.1	CT	2 Bytes Value
9	Single Touch- 2 Bytes	O	DPT1.1	CT	2 Bytes Value
10	Long Touch – 2 Bytes	O	DPT1.1	CT	2 Bytes Value
10	Neg Edge - 2 Bytes	O	DPT1.1	CT	2 Bytes Value
11	Short Touch – Toggle Value – 1 Bit	O	DPT1.1	WCTUI	Value to be toggled
11	Pos Edge – Toggle Value – 1 Bit	O	DPT1.1	WCTUI	Value to be toggled
11	Single Touch – Toggle Value – 1 Bit	O	DPT1.1	WCTUI	Value to be toggled
12	Long Touch – Toggle Value – 1 Bit	O	DPT1.1	WCTUI	Value to be toggled
12	Neg Edge – Toggle Value – 1 Bit	O	DPT1.1	WCTUI	Value to be toggled

The objects of the second button start at 21. There is a spacing of 17 objects between each button channel.

Dimming

You can use the dimming function for the following functions:
– dim up and down via one button (single-surface dimming)
– either dim up or down. You need a second button to dim in the other direction (dual-surface dimming).

You can use the corresponding button to switch the light on or off (short touch) or dim it using a longer touch. When switching takes place, an ON/OFF telegram is sent via the switch object. For single surface dimming, the value of this telegram is the opposite value of it's internal state.
When dimming, dimming up or dimming down is carried out via the Dim Relative object. In addition, you can also transmit the corresponding dimming step cyclically. For single surface dimming, the direction of the dimming is the opposite direction of its internal state.

List of Communication Objects and parameters:

Name	***Value Range***	***Comment***
Dimming Direction	Up / Down / Up And Down	direction of the dimming
Dimming Step	100%, 50%, 25%,12.5%,6.25%, 3%, 1.5%	dimming step
Long Touch Time (x 100ms)	3 – 50	minimum time a button must be touched to generate a long touch event
Cyclic Dimming	Yes/No	Enable cyclic dimming, send the dimming step at every interval
Cycle Time (x 100ms)	5 – 50	interval to send dimming step during cyclic dimming

No	***Name***	***I/O***	***DPT***	***Flags***	***Use***
4	Dimming – Switch	O	DPT1.1	CT	Switch Object
11	Dimming – Relative	I/O	DPT3.7	WCTUI	Relative Dimming Object
12	Dimming – Toggle Value	I	DPT1.1	WCTUI	Status value to Toggle (only for up/down dimming)

Blind

With the blind control function, you can operate blinds using one or two touch surfaces. A long touch initiates a long motion. After a short touch a step/stop telegram is sent. Using two buttons, the direction will always be the same, upwards or downwards, either when adjusting the slats (short touch) or lowering/raising the blinds (long touch).

When only one direction is chosen, the option "Send fixed Position after extra keypress" becomes available. It enables an extra communication object "Blind Fixed Position", which will send "up" or "down" when the button is touched within 1,5 seconds after a long motion was started. This event can be used to drive the blinds to a predetermined position in the actor.

When using one button for both directions, the direction of lowering/raising depends on the previous action. I.e. when the blind has just been moved downwards, it will move upwards the next time the button is touched for a long period.

During adjustment of the slats the direction is only changed after the "Slat Direction Reversal Time" has elapsed. After a stop/step telegram has been transmitted to adjust the slats, a stop/step telegram for the same direction can be created by touching the button again, as long as this subsequent push-button action is carried out within a time period, specified by the Slat Direction Reversal Time. If this time period has elapsed, the direction of rotation of the slats will change when the button is touched shortly.

List of the parameters and communication objects:

Name	***Value Range***	***Comment***
Blind Direction	Up / Down / Up And Down	direction of the blind
Long Touch Time (x 100ms)	3 – 50	minimum time a button must be touched to generate a long touch event
Slat Direction Reversal Time(x 100ms)	3 – 50	Minimum time between two subsequent touches during step(slat adjustment) to change the direction
Send fixed Position after extra keypress	Yes/No	Sends extra Up/Down command with keypress within 1,5 seconds after long motion was started

No	***Name***	***I/O***	***DPT***	***Flags***	***Use***
4	Blind – Step/Stop	I/O	DPT1.7	CT	Step/Stop (Short Motion) Object
5	Blind – Up/Down	I/O	DPT1.8	WCTUI	Up/Down (Long Motion) Object
6	Blind – Fixed Position	I/O	DPT1.8	CT	Up/Down (Long Motion) Object

Shutter

Shutters do not have any rotating slats, so the step object is omitted. There are basically 3 operation concepts to use for controlling shutters:

1. Start motion on long touch, stop on short touch (comparable to blinds).
2. Start motion on short touch, stop on long touch.
3. Single Touch for Starting/Stopping the shutters. Depending on the current state of the shutter actuator, the Intensity will stop or start the shutter when the button is touched. Holding the key while the shutter was active will start a movement in the opposite direction.

For Single Touch it is necessary to keep track of the motion, and thus the shutter motion time should be set to the duration of the shutters motion time, to go from entirely down to completely up.

However, when possible, it is better to use the motion status object, which is sometimes available on shutter actors. In this case the motion time is ignored.

It is necessary to set this accurately, otherwise the button might generate a new long motion if the key is being touched while the shutter is moving, or send a stop object while the shutter is actually not moving any longer.

List of the parameters and communication objects:

Name	***Value Range***	***Comment***
Shutter Direction	Up / Down / Up And Down	direction of the shutter with this button
Operation Concept	long: start – short: stop / short: start – long: stop / Single Touch	Behaviour of the shutter upon touch of the button
Long Touch Time (x 100ms)	3 – 50	minimum time a button must be touched to generate a long touch event
Shutter Motion Time(x 1s)	1 – 65	Time of movement of a shutter from completely down to completely up

No	***Name***	***I/O***	***DPT***	***Flags***	***Use***
4	Shutter Stop	I/O	DPT1.7	WCT	Stop Object
5	Shutter Up/Down	I/O	DPT1.8	WCTUI	Up/Down (Long Motion) Object
12	Shutter Motion Status	I	DPT1.10	WCTUI	Motion Status

Scene

A scene can be recalled by a short touch. If you also want to enable saving scenes, you can use the save function. Then a save telegram will be send out on a long key touch, after which the corresponding scene module starts saving the values of the actuators that belong to that scene. The period for the long touch time can be set.

If you need a more dynamic behavior of the scene, e.g. setting lights 100% at day, but only 10% at night, you can use the "Use scene number from external object". Some external logic determines beforehand which scene to recall when the touch surface is touched.

Examples of use:

- If a light is on in the room/house, touch this button to turn everything off in the room/house, otherwise turn on the light in that room.
- If the button is touched after midnight, the lights are set at 10%. Otherwise the light level is set to 100%.

List of the parameters and communication objects:

Name	***Value Range***	***Comment***
Use scene number from external object	Yes/No	whether to use a fixed scene value, or use one from an external communication object
Scene number	1 – 64	scene to be recalled when the button is touched
Use save function	Yes/No	long keypress will send out a save telegram for the current scene (from external object or fixed, as programmed)

No	***Name***	***I/O***	***DPT***	***Flags***	***Use***
6	Scene Value	O	DPT18.001	CT	Scene number
15	Scene Input Value	I	DPT18.001	WCTUI	Input object of scene number

LED control

The integrated LEDs provide status and tactile feedback. It is possible to control the color and brightness of the LED in many ways.

1. Always Off

2. Always on: color to be selected from a list of available colors, e.g. White During operation, this color will be fixed.

3. Bound to a status object, e.g. 1 = green, 0 = red

4. Upon button activation; distinct color when no touch detected, when touched, or when long touch time has been reached.

5. Based on External brightness object and color object. Brightness is an external scaling object that allows you to dim the LED. By filling in a number from 0 to 10 you can set the color List of Colors:

   0 = Off
   1 = White
   2 = Red
   3 = Green
   4 = Blue
   5 = Cyan
   6 = Magenta
   7 = Yellow
   8 = Violet
   9 = Orange

   The brightness object is a scaling object which will override the default LED brightness.

6. Through external RGB object(s). It's possible to control the color mixing of the multicolor LED over the bus by using the 3 R,G,B color components as a scaling object, or through the KNX specified RGB color object (DPT 232.600). General brightness parameters will be ignored.

In order to provide tactile feedback, (at most) 30 % of dim value (up to 100%) will be added to the current value so the LED lights up upon touch.

List of the parameters and communication objects:

Name	***Value Range***	***Comment***
LED Configuration	Disabled / Always On / Button Activation / External Bit Object / External Brightness and Color Byte / External RGB Object	mode of controlling the color and/or brightness of the LED
LED Color	White / Red / Green / Blue / Cyan / Magenta/ Yellow / Violet / Orange	list of predefined colors
LED Color when no Touch	Off / White / Red / Green / Blue / Cyan / Magenta/ Yellow / Violet / Orange	list of predefined colors
LED Color on Touch	Off / White / Red / Green / Blue / Cyan / Magenta/ Yellow / Violet / Orange	list of predefined colors
LED Color on Long Touch	Off / White / Red / Green / Blue / Cyan / Magenta/ Yellow / Violet / Orange	list of predefined colors
LED Color on Status = 0	Off / White / Red / Green / Blue / Cyan / Magenta/ Yellow / Violet / Orange	list of predefined colors
LED Color on Status = 1	Off / White / Red / Green / Blue / Cyan / Magenta/ Yellow / Violet / Orange	list of predefined colors
Default Color	Off / White / Red / Green / Blue / Cyan / Magenta/ Yellow / Violet / Orange	default color at startup. If ROI flag is enabled on the color object, this color will be overwritten as soon as the read response is received
Default Brightness	Off, 10-100%	default brigthness at startup. If ROI flag is enabled on the brightness object, this brightness will be overwritten as soon as the read response is received
Use 3 Bytes Color Byte	Yes/No	wheter you want to use a 3 Bytes DPT or 3 individual 1 Byte objects for each color component

No	***Name***	***I/O***	***DPT***	***Flags***	***Use***
16	LED control 1 Bit	I	DPT1.2	WCTUI	External Status Bit Object
17	LED control 3 Bytes	I	DPT232.600	WCTUI	External 3 Bytes Color Object
17	LED control – Red	I	DPT5.1	WCTUI	External Color Byte Object, scaling 0 – 100%
18	LED control – Green	I	DPT5.1	WCTUI	External Color Byte Object, scaling 0 – 100%
19	LED control – Blue	I	DPT5.1	WCTUI	External Color Byte Object, scaling 0 – 100%
19	LED control Color	I	DPT18.001	WCTUI	Color, 0 = Off / 1 = White / 2 = Red / 3 = Green / 4 = Blue / 5 = Cyan / 6 = Magenta/ 7 = Yellow / 8 = Violet / 9 = Orange
20	LED control Brightness	I	DPT1.2	WCTUI	External Brightness, scaling 0 – 100%

Temperature

All temperatures are in °C.

Internal Sensor

The Intensity contains an internal Temperature sensor which is sampled every 5 seconds. It measures temperatures from -40 °C to 125 °C. By using the parameter Temperature Correction you can correct the measured value when necessary. The temperature value can be send onto the bus after a cyclical time or when the new measured value differs too much from a previous value. The latter case will also reset the cyclical timer.

List of the parameters and communication objects:

Name	***Value Range***	***Comment***
Temperature Correction (x 0.1 °C)	-50 to 50	correction on measured value, in tenth of a degree (range -5° to +5°)
Send at a deviation of ... (x 0.1 °C)	0 to 50	send temperature automatically onto the bus as soon as the difference between the last sent temperature is bigger than the current temperature + or - this parameter. Set to 0 to disable
Send at least every ... minute(s)	0 to 60	send temperature periodically onto the bus. Set to 0 to disable.

No	***Name***	***I/O***	***DPT***	***Flags***	***Use***
2	Internal Temperature	O	DPT9.1	RCT	(Corrected) Internal Measured Temperature

Thermostat

You can use a built-in thermostat to control the climate of your room. Heating, Cooling, Heating + Cooling with manual or automatic switch is supported. If a wait time is specified, then the thermostat will wait that period after a switch occurred before controlling the heating/cooling.
The status of heating/cooling can be monitored by the heating/cooling mode feedback, or by the corresponding bits in the RHCC Status Feedback object.

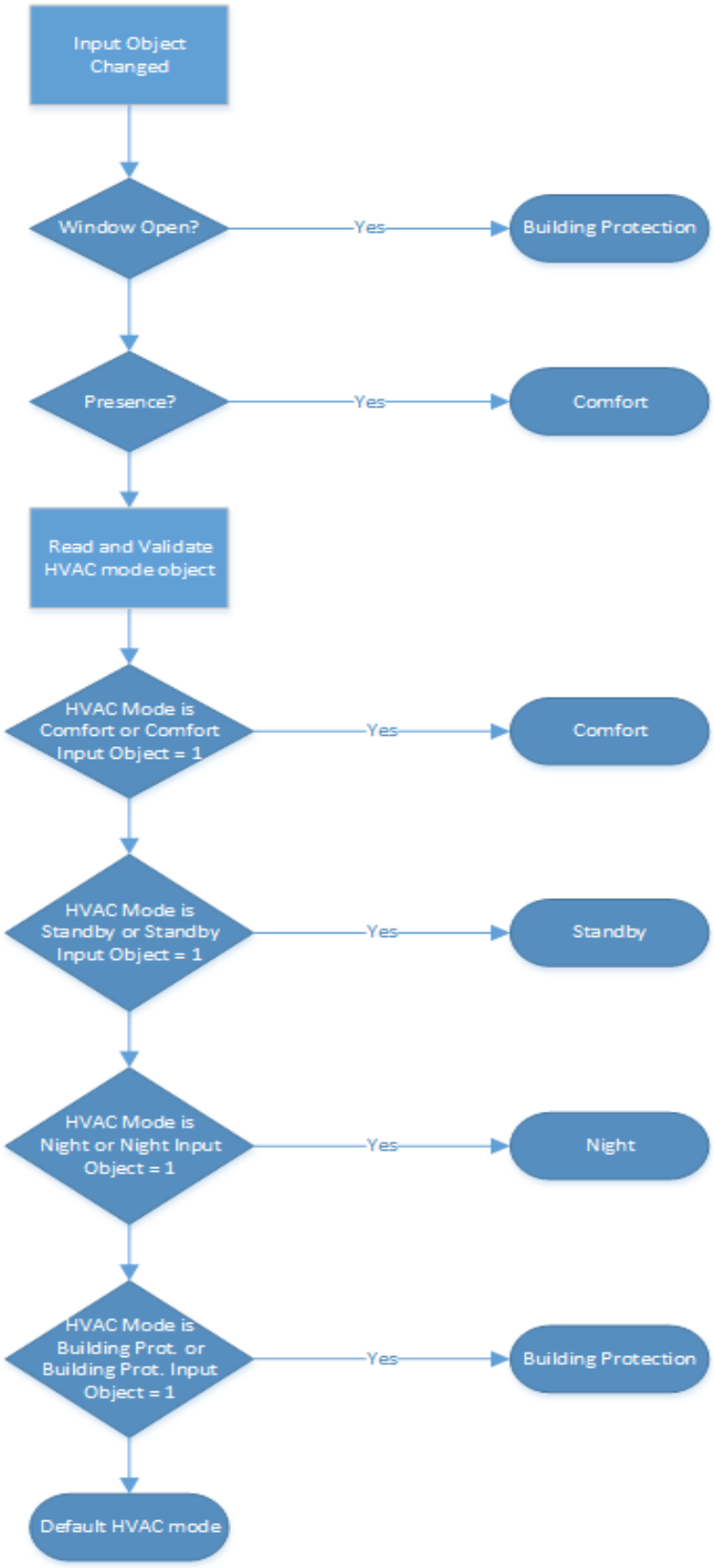
The method for controlling the climate can be 2-Step switching , PI continuous or PI switching. When switching is selected, an output object Heating (or Cooling) switch will appear. In the other case, a Heating (or Cooling) value will make you able to control your heating/cooling equipment.

4 Modes are supported :

- Comfort
- Standby
- Night
- Building Protection (a.k.a. Heat/Frost Protection)

The modes can be set over the HVAC object (DPT 20.102), or over the corresponding 1 Bit object. The Presence or Window open object can also alter the current mode.

For a schematic overview of how the current HVAC mode is determined, see below.


Input Object Changed
Window Open?
Yes
Building Protection
Presence?
Yes
Comfort
Read and Validate HVAC mode object
HVAC Mode is Comfort or Comfort Input Object = 1
Yes
Comfort
HVAC Mode is Standby or Standby Input Object = 1
Yes
Standby
HVAC Mode is Night or Night Input Object = 1
Yes
Night
HVAC Mode is Building Prot. or Building Prot. Input Object = 1
Yes
Building Protection
Default HVAC mode

Feedback about the current mode can be obtained through the mode feedback status objects or through the HVAC status object (DPT_HVACStatus).

Status and Errors

Through the RHCC Status Feedback object (DPT 22.101) you can track the status of the thermostat. Following bits are implemented:

Bit0 : Fault

Bit7 : Heating Disabled

Bit8 : HeatCool mode

Bit11 : Cooling Disabled

Bit12 : DewPoint status

Bit13 : FrostAlarm

Bit14 : Overheat Alarm

Bit 6 (Controller Status) of the HVAC Status object indicates whether the thermostat is actively heating/cooling.

The "Dewing Point Alarm" will disable the cooling immediately.

Frost Alarm will be set if the temperature drops below the frost protection setpoint. Overheat alarm when temperature is higher than the heat protection temperature.

Setpoint

The setpoint can be controlled in 2 ways:

- By writing a new temperature to the Setpoint object.
- Using the Setpoint Adjustment object. This can either be a floating point offset or a 1 bit switch object. The latter you can use to change the setpoint using button input. Writing a 0(up) will increase the offset with 0.5K, writing a 1 (Down) will decrease it.

You can also limit the values that are written directly to the setpoint object, by setting "New Setpoint within Adjustment bounds" to "Yes".

Feedback of the setpoint, or the adjustment, will be given through the feedback objects.

Actual Temperature

It's possible to use an external temperature sensor to determine the actual temperature of the room, optionally for a certain proportion. Feedback on the calculated temperature can be obtained through the "Actual Temperature Feedback" object.

Name	***Value Range***	***Comment***
Control Mode	Heating / Cooling / Heating and Cooling (Automatic Switch) / Heating and Cooling (Manual Switch)	Heating or Cooling Mode of the thermostat. When using the manual switch you can use the heating/cooling Selection object to set the desired mode(0 = Cooling, 1 = Heating). When the switch is done automatically, then the current mode will depend on the current temperature. When lower than the heating setpoint, heating mode will be activated. Higher than the cooling setpoint will set cooling mode active.
Wait Time after switch (min)	0 – 240	time to wait after a heating/cooling mode switch has occurred to actually control the HVAC equipment.
Default HVAC Mode	Comfort / Standby / Night / Building Protection	when the current HVAC mode cannot be determined (when the other objects have not be written or at 0), the thermostat will switch to this mode
Use external Temperature Sensor	Yes/No	whether you want to use an external temperature sensor or the internal one. The actual temperature the termostat is using can be read from the Actual Temperature Feedback object
Proportion external sensor	20 % / 40 % / 50 % / 60 % / 80 % / 100 %	proportion of the external sensor that is used to calculate the actual temperature
Maximum Adjustment up	0°C to 7°C	maximum upward offset that can be set on the Setpoint Adjustment Object
Maximum Adjustment down	0°C to 7°C	maximum downward offset that can be set on the Setpoint Adjustment Object
Setpoint Adjustment Over	2 Byte Floating Point Object / 1 bit Object	wheter you want to use a 1 bit object (0 = +0.5°, 1 = -0.5°) or a floating point object to set the offset
Setpoint Comfort Mode	5°C – 40 °C	Setpoint in HVAC mode “Comfort”
Standby Offset	+/- 0°C - 7°C	Offset applied to Setpoint in Standby Mode
Night Offset	+/- 0°C - 7°C	Offset applied to Setpoint in Night Mode
Setpoint Frost/Heat Protection	5°C – 40 °C	Setpoint in HVAC mode “Building Protection”
Control Method	2-Step Switching / PI Switching / PI Continuous	method to determine the heating/cooling. The best method depends on the type of HVAC equipment
Hysteresis Up	0.3 °C to 2.0 °C	the difference the Setpoint should surpass to stop heating/cooling
Hysteresis Up	0.3 °C to 2.0 °C	the difference the Setpoint should surpass to start heating/cooling
PWM Cycle Time (minutes)	1 to 60 minutes	the period of time of the PWM cycle duration when PI Switching is selected
Cooling System	Cooling Ceiling (5K / 240 min) / Fan Air Convector (4K / 90 min) / Split Unit (4K / 90 min) / User Defined	the type of cooling system. The differential and proportial factors for the PI controller are derived from this
Heating System	Warm Water Heating (5K / 150 min) / Underfloor Heating (5K / 240 min) / Electric Heating (4K / 100 min) / Fan Convector (4K / 90 min) / Split Unit (4K / 90 min) / User Defined	the type of heating system. The differential and proportial factors for the PI controller are derived from this.
Proportional Range (x 0.1K)	10 – 50	the proportional factor of the PI controller
Reset Time (minutes)	0 – 240	the reset time of the PI controller

No	Name	I/O	DPT	Flags	Use
157	External Temperature	I	DPT9.1	WCTUI	Temperature from external sensor
158	Setpoint	I	DPT9.1	WCTUI	Current Setpoint
159	Setpoint Adjustment	I	DPT9.1	WC	Setpoint adjustment value (float value)
160	Setpoint Adjustment	I	DPT1.8	WC	Setpoint adjustment value (1 Bit input – Up/Down)
161	Dewpoint Alarm	I	DPT1.2	WC	dewpoint alarm for Thermostat in Cooling Mode
162	Presence	I	DPT1.2	WC	Presence object for determing the HVAC mode. Normally HVAC mode will switch to Comfort mode
163	Window Open	I	DPT1.2	WC	Window Open object for determing the HVAC mode. Normally HVAC mode will switch to Building Protection mode
164	HVAC mode	I	DPT20.102	WC	HVAC mode object for controlling the HVAC mode according to the values defined in DPT_HVACMode [0 .. 4]
165	Frost/Heat protection Mode	I	DPT1.2	WC	switches the thermostat in Frost/Heat protection mode
166	Comfort Mode	I	DPT1.2	WC	switches the thermostat into comfort mode
167	Standby Mode	I	DPT1.2	WC	switches the thermostat into standby mode
168	Night Mode	I	DPT1.2	WC	switches the thermostat into night mode
169	Heating/Cooling selection	I	DPT1.2	WCTUI	if heating/cooling mode is set to switch manually, then writing 1 onto this object activates the heating mode
170	RHCC Status Feedback	O	DPT22.101	RCT	status information, bits defined according to DPT_RHCCStatus
171	Actual Temperature Feedback	O	DPT9.1	RCT	the calculated temperature, from internal and external sensors, used by the thermostat
172	Current Setpoint Feedback	O	DPT9.1	RCT	the actual setpoint, adjustment included
173	Setpoint Adjustment Feedback	O	DPT9.1	RCT	the adjustment applied
174	HVAC Status Feedback	O	--	RCT	HVAC mode feedback, bits according to DPT_HVACStatus
175	Frost/Heat Protection Feedback	O	DPT1.2	RCT	HVAC mode feedback, whether thermostat is in Frost/Heat Protection mode
176	Comfort Mode Feedback	O	DPT1.2	RCT	HVAC mode feedback, whether thermostat is in Comfort mode
177	Standby Mode Feedback	O	DPT1.2	RCT	HVAC mode feedback, whether thermostat is in Standby mode
178	Night Mode Feedback	O	DPT1.2	RCT	HVAC mode feedback, whether thermostat is in Night mode
179	Cooling Switch	O	DPT1.1	CT	Cooling object, active when the cooling is actively cooling the room.
180	Cooling Value	O	DPT5.1	CT	The calculated PWM value from the PI controller.
181	Heating Switch	O	DPT1.1	CT	Heating object, active when the thermostat is actively heating the room.
182	Heating Value	O	DPT5.1	CT	The calculated PWM value from the PI controller.
183	Cooling Mode feedback	O	DPT1.2	RCT	feedback whether the thermostat is in cooling mode
184	Heating Mode feedback	O	DPT1.2	RCT	feedback whether the thermostat is in heating mode

Modules

Under modules you can activate additional functionality that comes with each switch.

- Scene Module, supporting eight scenes with eight actuators (1 Bit/1Byte/2Byte supported)
- Basic Logic Module (AND/OR functions), consisting of five Logic Channels that each have up to five 1-Bit inputs
- Timers, up to four
- Up/Down Counters, up to four

Scene Module

The scene module is a matrix of 8 actuator groups, with 8 scenes that **optionally** have a value for every actuator. If you want an actuator not to change with a scene, you can specify not to use this value for this scene. For actuator 1-6 the type must be 1 Bit or 1 Byte. Actuator 7 and 8 can additionally contain a 2 Byte value.

It is also possible to save scenes. When the scene module receives a request to save a scene, it will issue a read request for the corresponding actuators and wait 1 second to receive all the read responses. It then saves and start using the received values.

It's also possible to choose not to overwrite the existing scene parameters. This is useful in case the end user already changed the scene their selves using the scene save functionality after a long touch. If you change the type of an actuator, then you must set this parameter to "No".

List of Parameters and Communication objects:

Name	***Value Range***	***Comment***
Overwrite Existing Scenes	Yes/No	Overwrite the scenes that were specified in the past. Otherwise, use the scene values from the parameters
Actuator Type Group *1-6*	Switch 1 bit / Value 1 Byte	Value type of actuator
Actuator Type Group *7-8*	Switch 1 bit / Value 1 Byte / Value 2 Byte	Value type of actuator
Scene number	0 – 63	Number of this Scene. Writing this value to the com object "Scene Function" will activate this scene
Use Value *X*	Yes/No	whether to sent the value to the actuator in this scene, or ignore it
Value	On / Off	value for 1 bit actuator
Value	0-255	value for 1 Byte actuator
Value	0-65535	value for 2 Byte actuator

No	Name	I/O	DPT	Flags	Use
185	Scene Function	I	DPT18.1	WC	Input object of scene number of type DPT_SceneControl
186	Actuator 1 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
187	Actuator 1 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
188	Actuator 2 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
189	Actuator 2 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
190	Actuator 3 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
191	Actuator 3 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
192	Actuator 4 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
193	Actuator 4 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
194	Actuator 5 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
195	Actuator 5 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
196	Actuator 6 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
197	Actuator 6 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
198	Actuator 7 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
199	Actuator 7 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
200	Actuator 7 – Value 2 Byte	I/O	DPT7.1	WCTU	2 Byte value to be sent / saved when a scene is recalled / saved
201	Actuator 8 – Switch 1 Bit	I/O	DPT1.1	WCTU	1 bit value to be sent / saved when a scene is recalled / saved
202	Actuator 8 – Value 1 Byte	I/O	DPT5.10	WCTU	1 Byte value to be sent / saved when a scene is recalled / saved
203	Actuator 8 – Value 2 Byte	I/O	DPT7.1	WCTU	2 Byte value to be sent / saved when a scene is recalled / saved

Timer

A Timer object can be used to start an action after another one has occurred, with a delay time. It is also possible to send out a value cyclically. In the latter case, a value will be sent each time the timer expires, as long as the input Activation Object is 1.

The timer period is Factor x Time Base, allowing you to specify a period from 100 milliseconds up to 255 hours.

You can choose when to activate the timer; whether a 1, or a 0 is written to the object, or on both edges. There is no way to cancel a non-cyclical timer that has been activated.

The value that is sent can be freely chosen. The copy/invert of a 1 Bit object will use the "Input Value" when the timer expires, not the value at the time of activation of the timer.

If a timer is "Resettable", then an incoming telegram on the activation object will reset the timer period to 0.

List of parameters and communication objects:

Name	***Value Range***	***Comment***
Time Base	100 milliseconds / 1 second / 1 minute / 1 hour	base of time calculation
Time Factor	1 – 255	multiplied by Time Base to get the timer Period
Value	On / Off / Invert / Copy	Value to be transmitted. When Invert or Copy is selected than an 1 bit input object will be available onto which the input value must be written so that the desired operation can be carried out.
Value	0 – 255	1 Byte Value to be transmitted after timer expiry
Value	0 – 65535	2 Byte Value to be transmitted after timer expiry
Use Value from External Object	Yes/No	Available when 1 / 2 Byte output value type is selected. Provides a means to sent out a copy of a value when the timer expires.
Cyclic	Yes/No	timer is cyclic, thus will be restarted automatically every time the timer expires. When the activation object is set to 0, then the timer will stop.
Use Value from External Object	Yes/No	Available when 1 / 2 Byte output value type is selected. Provides a means to sent out a copy of a value when the timer expires.
Activation on	On / Off / Both Edges	Edge on which the timer has to start
Resetable	Yes/No	a new write onto the input object that matches the Activation Parameters will reset the timer

These are the communication objects for Timer 1. Timer 2 starts at 248, with 5 objects intermittently for subsequent timers.

No	*Name*	*I/O*	*DPT*	*Flags*	*Use*
241	Activate	I	DPT1.1	WC	Activate the timer
242	Input Value – 1 Bit	I	DPT1.1	WC	Input value of a 1 Bit timer
242	Input Value – 1 Byte	I	DPT5.10	WCTUI	Input value of a 1 Byte timer
242	Input Value – 2 Byte	I	DPT7.1	WCTUI	Input value of a 2 Byte timer
243	Switch 1 Bit	O	DPT1.1	CT	Output value of a 1 Bit timer
244	Output Value – 1 Byte	O	DPT5.10	CT	Output value of a 1 Byte timer
245	Output Value – 2 Byte	O	DPT7.1	CT	Output value of a 2 Byte timer

Up/Down Counter

The Up/Down counter makes it possible to maintain a 1Byte unsigned value centrally, which can be increased/decreased with the specified step value by writing a 1(decrease)/0(increase) onto the input object. The bounds of this value will limit its range. It is also possible to reset the value by writing a 1 to the Reset Object.

List of parameters and communication objects:

Name	*Value Range*	*Comment*
Reset Value	0-255	initial value, or value to be set when the reset object is set to 1
Step Value	0-255	value to be added/subtracted each time the input object is se
Minimum Value	0-255	the minimum value the counter can have. The counter will stop subtracting values once this value has been reached
Maximum Value	0-255	the maximum value the counter can have. The counter will stop adding values once this value has been reached

These are the communication objects for Up/Down Counter 1. Object for subsequent Up/Down Counters have 3 objects intermittently.

No	*Name*	*I/O*	*DPT*	*Flags*	*Use*
261	Input Value	I	DPT1.8	WC	Input value , 0 (Up) from adding the step value to the counter, 1(Down) to subtract it.
262	Reset	I	DPT1.1	WC	Input value
263	Output Value	O	DPT5.10	CT	Output value of the counter

Logic Module

Up to 5 logic channels can be defined. Each channel has up to 5 logic inputs, which can be inverted individually. You can use those to make an AND/OR comparison. The result of the function will be set onto the KNX bus, depending on the chosen setting:

1. Only when the result of the logic function changes.

2. Every time something is written onto an input object.

4 Output types can be selected: Switch 1 bit, 1 Byte, 2 Byte and 3 Byte RGB object (DPT232.600).You can also use this mechanism as a value converter, enabling you to generate a value from the above types, starting from a single 1 bit communication object.

At startup the initial values of the logic inputs (0 by default, 1 if they are inverted) will be evaluated and the result of the function will always be sent onto the bus. Enable the ROI flag if you want to effectively read the object's value at startup. Upon reception of the value the logic function will be executed.

List of parameters and communication objects:

Name	***Value Range***	***Comment***
Logic Function	And / Or	type of logic function to apply
Number of Input Objects	1 – 5	how many 1 bit inputs the function uses
Invert Input 1	Yes/No	whether to invert input 1
Invert Input 2	Yes/No	whether to invert input 2
Invert Input 3	Yes/No	whether to invert input 3
Invert Input 4	Yes/No	whether to invert input 4
Invert Input 5	Yes/No	whether to invert input 5
Sending Condition	Not Automatic / When Input Object is Written / When Result Changes	when to send the result of the logic function
Output Value Type	Switch 1 bit / Value 1 Byte / Value 2 Byte / Value 3 Byte	value type of the result
Send value when expression is True	Yes/No	whether a value is to be sent when the expression evaluates to True
Send value when expression is False	Yes/No	whether a value is to be sent when the expression evaluates to True
Value	On / Off	1 Bit result
Value	0 – 255	1 Byte result
Value	0 – 65535	2 Byte result
Value Byte 1	0 – 255	1 Byte part when value Type is 3 Byte. This byte corresponds to Red when using DPT232.600 (RGB value)
Value Byte 2	0 – 255	1 Byte part when value Type is 3 Byte. This byte corresponds to Green when using DPT232.600 (RGB value)
Value Byte 3	0 – 255	1 Byte part when value Type is 3 Byte. This byte corresponds to Blue when using DPT232.600 (RGB value)

List of communication objects for Logic Function 1.

No	***Name***	***I/O***	***DPT***	***Flags***	***Use***
211	Input 1	I	DPT1.2	WC	Logic Input Object 1
212	Input 2	I	DPT1.2	WC	Logic Input Object 2
213	Input 3	I	DPT1.2	WC	Logic Input Object 3
214	Input 4	I	DPT1.2	WC	Logic Input Object 4
215	Input 5	I	DPT1.2	WC	Logic Input Object 5
216	Switch – 1 Bit	O	DPT1.1	CT	1 Bit Switch Output Object
216	Value – 1 Byte	O	DPT5.1	CT	1Byte Output value
216	Value – 2 Byte	O	DPT7.1	CT	2Byte Output value
216	Value – 3 Byte	O	DPT232.600	CT	3Byte Output value