SAIA BURGESS CONTROLS

PCD7.L611 room controllers LON

Document 26/876; Edition EN03 | 2014-04-07

0

0 Contents

0

0.1 Document History

Date	Version	Changes	Remarks
2009-08-27	EN01	-	Adaptation of Comtec documentation
2010-05-10	EN01	-	publication
2013-09-30	EN02	-	New Logo and new company name
2014-05-07	EN03	-	diverse

0.2 About this manual

See the section in the appendix in relation to some of the terms, abbreviations and the references used in this manual.

This manual and the books mentioned in the notes are not sufficient for a successful Lon configuration. They serve only to basic education. The training for the LON Certified System Integrator is offered by the LonMark country-specific organizations.

Each country has its own Lon Organization (LonMark) for training of system integrators and certificates.

LonMark international : http://www.lonmark.org

Country-specific such. : http://www.lonmark.de

0.3 Brands and trademarks

Saia PCD® and Saia PG5®
are registered trademarks of Saia-Burgess Controls AG.

Technical modifications are based on the current state-of-the-art technology.

Published in Switzerland

1 Overview

1.1 Room automation solution with SBC Serial S-Net or LonWorks®

The PCD7.L6xx room controllers, based on SBC Serial S-Net, LonWorks® or BACnet® MS/TP networks, are mainly used for HeaVAC applications with FanCoil devices, radiator/cooled ceiling combinations or VVS systems. The extension module for light and shade allows the electrical systems to be easily integrated in to the room automation solution. Customer-specific operating concepts can be produced with the wide range of room control units. these room control units are connected to the room controller by cable, infra-red or wireless receivers.

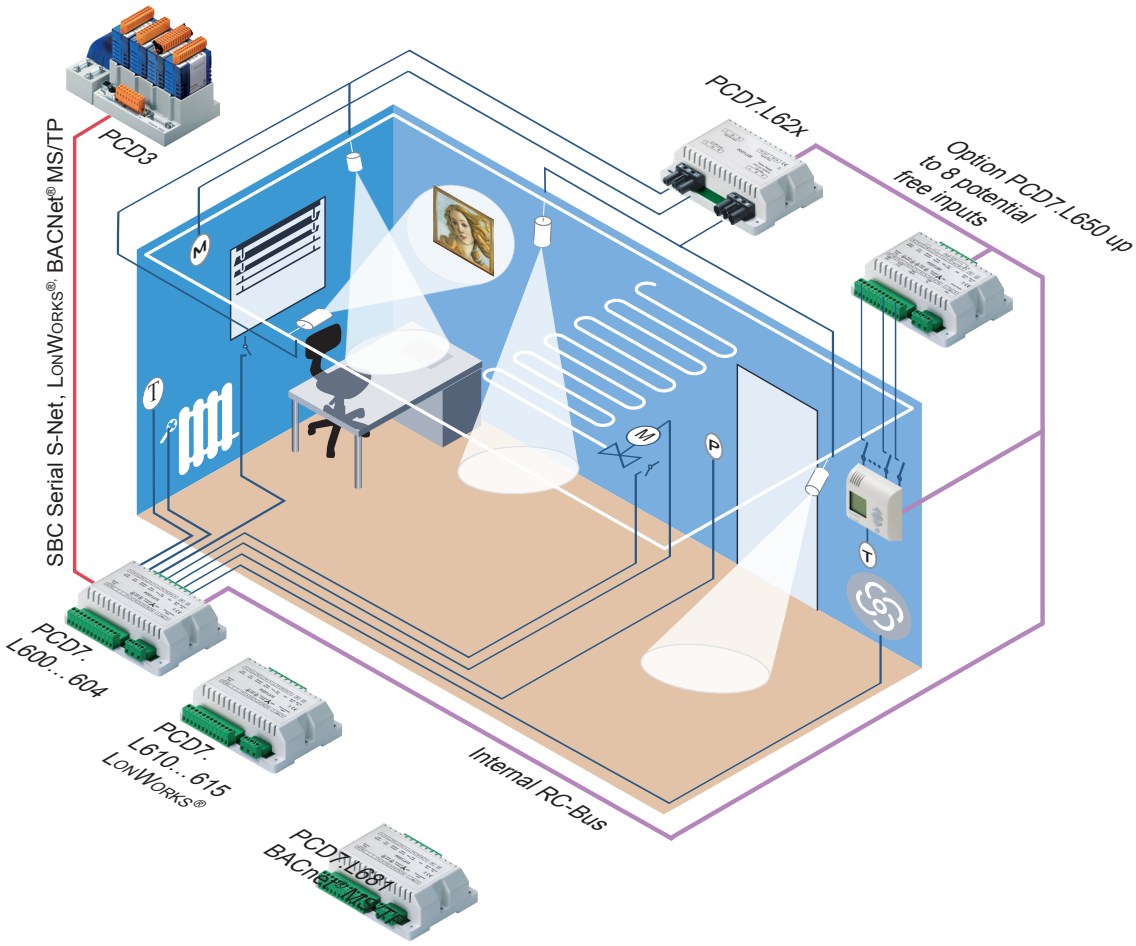
Manufacturer-independent room control units

Control units with LonWorks® communication can be connected directly to the LON room controllers. To connect EnOcean room components there is a receiver module that can be connected directly to the room controller via the internal RC bus. If the user control requirements should still not be met in terms of form, design or functionality, the system integrator can use the open interfaces to the automation station or analogue room control units to combine the room controller with third-party systems.

1

Features:

- Wide range of uses with parameter-driven application programs
- Room controllers for communication via SBC Serial S-Net, LonWorks® or BACnet® MS/TP*
- Expansion modules for electrical systems
- Wide range of analogue, digital and mobile room control units
- Options to combine the basic controller with room control units from third-party providers



* in preparation

1

1.2 Possible uses for the PCD7.L6xx series

1.2.1 Standalone control with no communication

The controller regulates the room temperature without any connection to a bus system. Control is handled entirely by the individual room controller based on the specified default parameter settings.

The outputs are driven by a control algorithm depending on the measured temperature.

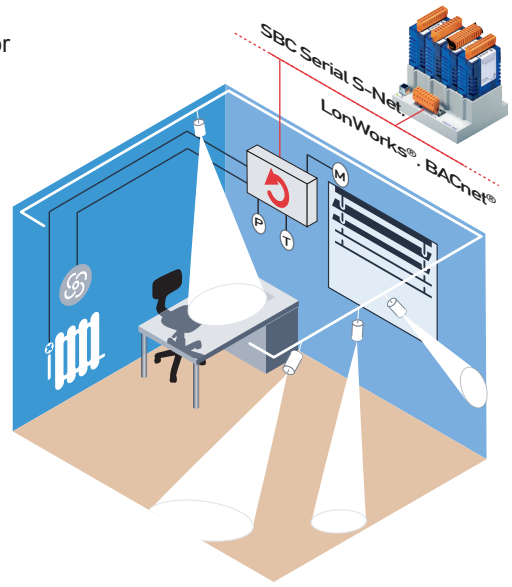
The default set-point setting of 21 °C can be modified by the set-point control (according to the device).

1.2.2 Standalone control with communication to the automation station

The controller is run as a slave station with a unique Bus address within a SBC Serial S-Net, LonWorks® or BACnet® network. Control is handled by the individual room controller with its own control algorithm.

The control functions - time or event-driven - are passed to the individual room controller by the automation station via suitably configurable function objects or network variables. This supports individual parameterisation and operation of the room controller. The device, and hence the control function, can also be influenced at any time via the Saia PCD® master station.

For parameterisation, there is a function object available in the library for every room controller type. In the case of open network connections, this is handled via network variables or network objects.



1.2.3 External regulation and control via the automation station

The Saia PCD® master station handles all regulation and control tasks. The room controller itself is only used as a remote input/output unit. Regulation and control can then be adapted to requirements in a very flexible way.

For parameterisation, RIO function objects are provided in the room controller library.

1.3 Application overview for the PCD7.L6xx series

Conformity table for PCD7.L61x range Name of the product PCD7.	.L610	.L611	.L614	.L615	.L616
Hardware					
Powersupply	230 VAC	230 VAC	230 VAC	230 VAC	230 VAC
PWM	2x 230 VAC	2x 230 VAC	2x 24VAC	4x 230 VAC	2x 230 VAC
0 - 10V	-	2x	2x with 24VAC supply	2x	2x
Fan 230V	3-step relay	3-step relay	3-step relay	2x 1-step relay	3-step relay
Electric heater (relay with pot. free contacts	1 relay	1 relay	1 relay	2 relay	1 relay
Applications					
Simple loop	X	X	X	X	X
Double loop	-	-	-	X	-
3 speeds fan	X	X	X	-	X
Variable speed fan	-	-	X	X	X
Frost guard mode	X	X	X	X	X
Air quality	-	-	X	-	X
Flow control	X	-	-	X	-
Blowing temperature limitation	X	X	X	-	X
Dew point	X	X	X	X	X
Direct control of outputs	X	-	X	-	X
Master/slave mode	X	X	X	X	X
Counting operation	-	-	X	-	X
Light	-	X	-	X	-
Shade	-	X	-	-	-

1.3.1 Operating modes

The 4 operating modes are set according to presence detection, the window contact and the instructions from the communication master

Comfort
Standard operating mode for when the room is occupied

Standby
Reduced operating mode used when the premises are temporarily unoccupied.

Reduced
Reduced operating mode when the premises are unoccupied for a long period of time.

Frost protection
The heating control is activated when the temperature drops below 8 °C (e.g. when a window is open)

1.3.2 Commissioning

When the room controller is used in a SBC S-Bus network, configuration is either by the Saia PCD® PCS Master, the Saia PG5® programming tool, or dedicated PC software. Practical function blocks (FBoxes) simplify commissioning.

Where the room controller is used within a LON network, the configuration is set via a LonWorks® plug-in.

The room controller satisfies the user profile "FAN Coil Unit Object (8020)"LonMark®.

1

1.3.3 Device overview and technical details of the room controller

SBC Serial S-Net	 PCD7.L600	 PCD7.L601	 PCD7.L603	 PCD7.L604 *
LonWorks®	 PCD7.L610	 PCD7.L611		 PCD7.L614 *  PCD7.L615 *
BACnet® MS/TP	 PCD7.L681 *			

Analogue inputs	Temperature sensor NTCA 010-040, Set-point potentiometer 10 kΩ | linear, 0…10 V			2 — —
Digital inputs	Main contact (e.g. window contact) Auxiliary contact selectable by user (e.g. presence, condensation, change-over…)			2 2
Analogue outputs	—	2×0…10 VDC		2
Digital outputs	2×Triac 230 VAC (10 mA…800 mA)	2×Triac 24 VAC (10 mA…800 mA)		4×Triac 230 VAC (10 mA…800 mA)
Relay outputs	3-step fan (4 connections) 230 VAC (3 A)			—
	Relays for electric heating: max. output 2 kW			2
Voltage supply	230 VAC with electronic fuse	24 VAC with electr. fuse	230 VAC with electr. fuse	
Current onsumption	approx. 100 mA			
Protection type	IP 20			
Dimensions	132 × 95 × 45 mm			
Temperature range	5…45 °C, 80% RH			
			The max. ouput power is 7 VA.	

Communication with SBC Serial S-Net	
Interface	RS-485, max. cable length 1200 m, 128 .L60x room controllers on one Saia PCD®
Transmission rate	Master, without repeater*
Protocol	4800, 9600, 19200, 38400, 115200 bit/s with automatic detection after restart SBC S-Bus data mode (slave)
Addressing at commissioning time via SBC S-Net or an external manual control device. Bus terminal resistors to be installed on site - integrated with L600, L601 and L604, software-activated	

Communication with LonWorks®	
Interface	FTT 10a
Transmission rate	78 kBit/s
Topology	Free topology max. 500 m; bus topology max. 2700 m
Number of LON nodes	max. 64 per segment, over 32000 in a domain/according to LonMark® 8020 profilet

Communication with BacNet® MS/TP	
Interface	RS-485, max. cable length 1200 m, 128 .L68x room controllers, without repeater*
Transmission rate	9600, 19200, 38400, 78600 bit/s - factory setting 38400 bit/s
Protocol	BacNet® MS/TP

* in mixed operation with RS-485 standard transceivers, note the minimum impedance
** in preparation

1.3.4 Phased-out room controllers

Item	*Active since*	*Not recommended for new projects*	*Phased out (production ceased) valid until / Commercial Info*
PCD7.L600	April 2007		
PCD7.L601	April 2007		
PCD7.L602			Aug. 2008
PCD7.L603	Sep. 2008		
PCD7.L604	June 2009		
PCD7.L610	April 2007		
PCD7.L611	April 2007		
PCD7.L614	June 2009		
PCD7.L615	June 2009		
PCD7.L681	2010		

2 Introductions

2.1 Lon networks characteristics

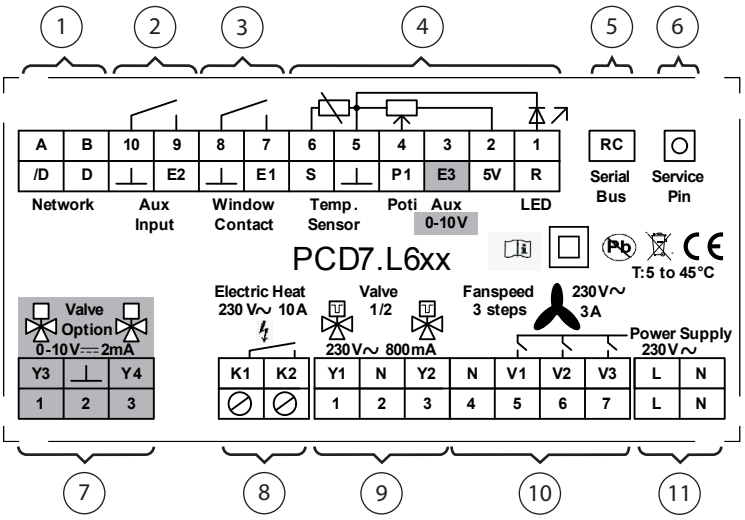
Program ID: 8F:FF:5B:55:01:04:04:70

Resource files: SBCScc with scope 5 – 8F:FF:5B:55:01:04:04:XX

Self documentation: PCD7L611 v101

2.2 Interface

	Desription
1	LON network
2	mixed input (NTC or contact) or (Aux contact)
3	input window contact (Main contact)
4	mixed inputs - (NTC or contact) or (Sensor) - Fan speed forcing - 5V output - LED operation status output
5	serial bus (RJ9 connectors, either for room operation unit or extension devices)
6	push button (service pin)
7	terminals 0V-10V outputs:
8	electric heater outputs 230 VAC / 10A
9	3 terminals for two 230 VAC valve outputs
10	4 terminals for three 230 VAC fan outputs
11	power input connector (230 VAC)



it's optional

3 Mounting instructions

3.1 Safety instructions

To guarantee safe operation, the PCD7.L6xx devices should only be operated by qualified staff according to the details given in the operating instructions and in compliance with the technical data. Qualified staff are people familiar with the assembly, commissioning and operation of the devices and suitably trained for their job.

When using the system, the legal and safety regulations applicable to the specific type of use must also be observed.

The room controllers have undergone a comprehensive pre-delivery inspection, ensuring that they left the factory in perfect condition.

Before commissioning, the devices should be checked for damage arising from incorrect transport or storage.

Removal of the identification numbers will invalidate the warranty.

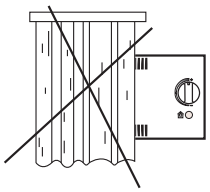
Please ensure that the limits specified in the technical data are not exceeded. Failure to do so may result in defects in the modules and the perpiherals connected to them. We can accept no responsibility for damage arising from improper deployment and use.

The plugs must never be inserted or removed with the power on. When installing or de-installing the modules, all components must be switched off.

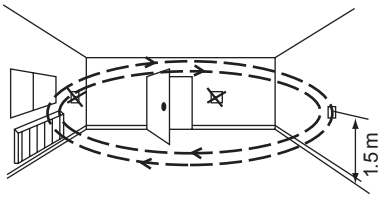
Please read this manual carefully before assembling and commissioning the modules. This manual contains instructions and warnings that must be observed to assure safe operation.

3.2 Assembly instructions

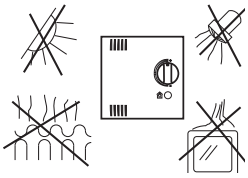
- The individual room controllers must only be installed and connected by an expert in accordance with the wiring diagram. Existing safety standards must also be observed.
- The individual room controller can only be used to regulate the temperature in dry, closed rooms. The maximum permissible relative humidity is 90%, non-condensing.
- Precise temperature measurement is subject to certain requirements as to the positioning of the temperature sensors. This applies both to the room control device itself and to the externally connected temperature sensor.
- The device can be mounted directly on the wall or flush-mounted within a pattress box.



Avoid direct exposure to sunlight or light from powerful lamps.



Do not install next to windows and doors because of draughts.



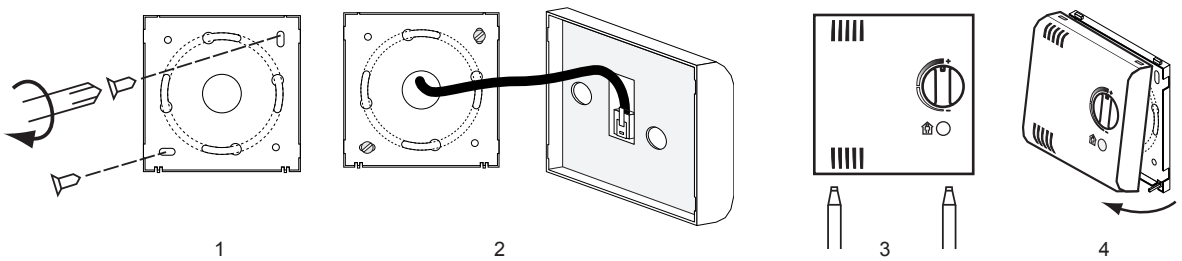
Do not install next to heat sources such as heaters, refrigerators, lamps etc.

Please ensure

- that all wires are screwed down tight
- that the connecting plug is properly engaged
- that the ventilation slots are placed above and below (positioning)
- that the device is mounted horizontally.

Mounting

Removal



1
Assembling the housing base

2
Wiring to the device

3
Unlocking

4

4 Functionalities

4.1 Functional Block Overview

- 1x **Node object** (see Ch 5.1)
- 1x **sccFanCoil** object (see Ch 5.2)
- 1x **AuxInput** object
- 1x **remoteCommand** object
- 4x **LampActuator** object
- 4x **SunblindActuator** objectt
- 1x **Virtual Functional Block** (see Ch 5.3)

For detail description see chapter 5

4.2 Inputs / Outputs configuration

All modifications on configuration variables are not consider immediately or on the next execution of the control process loop. It is highly recommended to restart the device after the complete configuration to be sure to active all new configurations. This can be done by unplug and plug again the power supply connector or by the network.

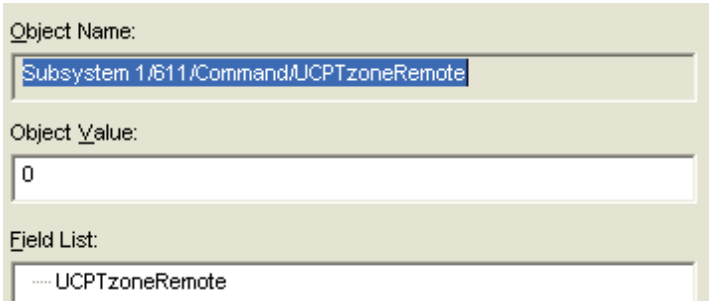
4.2.1 Room operation unit

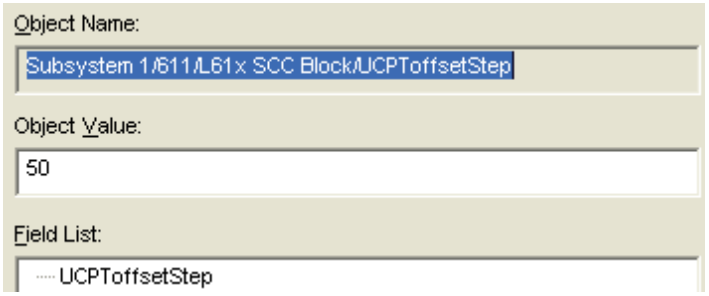
The PCD7.L611 can be used with a room local device to make available to the controller inputs needed for the regulation. The local device provides at the same time an interface for users to check and act on the process regulation (fitting of occupation, set point, ventilation…).

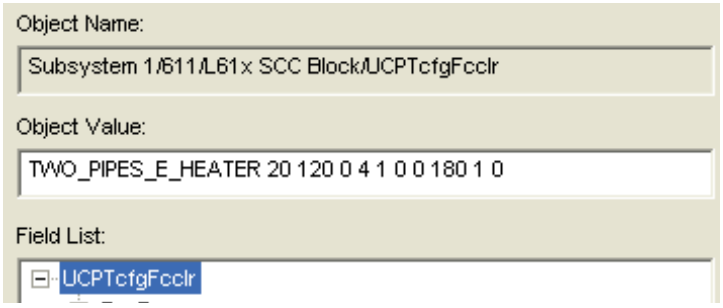
Remote operation unit used with the controller can be digital and plug on the “serial input”, or analogue and plug on standard inputs “S” to “R”. To learn more about these units, look at the document “Room controller unit PCD7.L61x, extension modules, accessories”.

HOW TO CONFIGURE THE ROOM OPERATION UNIT?

In the following description, only variables for room operation unit configuration are described.

nciZoneRemote	For remote operation unit, a zone address needs to be configured in each one to be sure to act on the correct room controller. This variable allows defining which number for the room operation unit can be takes in account by the PCD7.L610. Its value is bordered from 0 to 30. This setting doesn't serve to configure the zone address into the remote operation unit. It is used only to consider orders with a number which match to this variable. To configure the remote operation unit and its zone address, refer to its own documentation.
	
0:	Universal receiver. Accept each remote control unit, whatever its number.
X:	(from 1 to 30): Accept only orders and information from a remote control unit with the same zone address.

nciOffsetStep	Value of one step for the offset set point adjustment on the room operation unit. This value is in hundredth of °C and is bordered from 0 to 255.
	

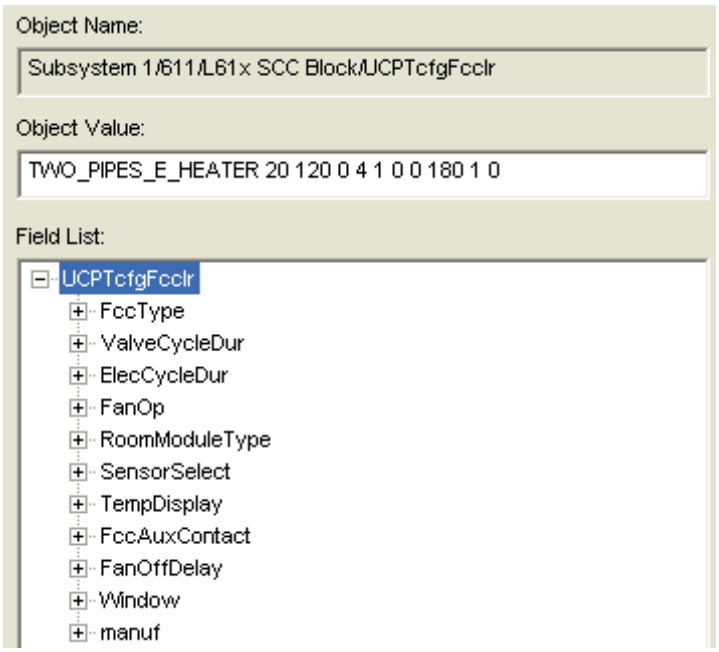
nciOffsetTemp	Value of the offset applied by default on the temperature sensor selected with the nciCfgFcc.sensorSelect (analogue or digital sensor). This value is in °C and is bordered from -10°C to 10°C.
	

4.2.2 Analogue Inputs

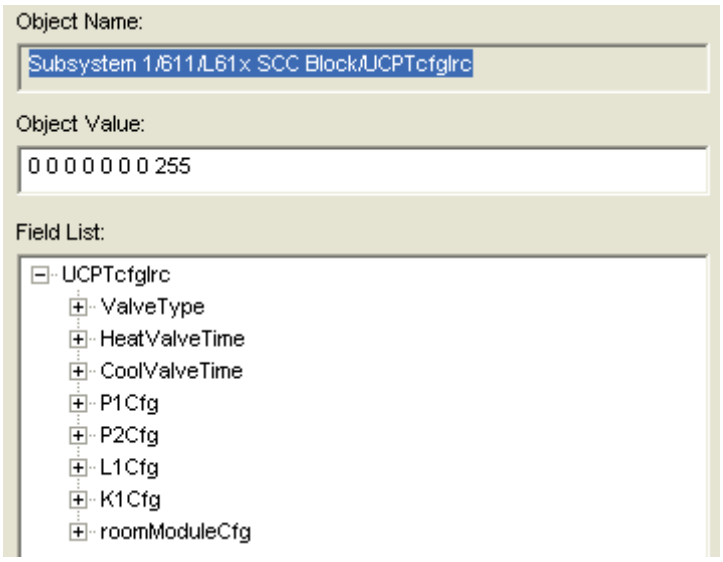
In software configuration properties, inputs are named as “input1” to “input6”. To make the conformity between these names and these wrote on the device hood, you can use this table which described the type of inputs (such as analogue, digital …).

Input	Pin	DIGITAL	NTC	Internal code
FccAuxContact	E2	X	X	Auxiliary contact, depend on its configuration
Window	E1	X		For window/door opening detection
/	S		X	Sensor input for room temperature
P1Cfg	P1	X	X	Set point adjustment
L1Cfg	R	X		LED output for analogue room operation unit or presence detector input

To use inputs P1Cfg and L1Cfg with their default functions, you need to configure the room operation unit as an analogue one (PCD7.L63x).

nciCfgFcc	Allows configurations for Aux input and window contact on the PCD7.L611.	
		
	.FccAuxContact	Configuration of function associated to the auxiliary input. Depending on its configuration, the input state can be displayed by the **33uxContact** or the **nvoAuxSensor**.
		2 — Change over state contact
		3 — Dew point detector
		5 — Temperature sensor
	.Window	Polarity configuration of the window contact.
		-1 — Always closed
		0 — Contact normally closed (NC).
		1 — Contact normally open (NO).

nciCfgIrc	Allows different configurations for the PCD7.L611, but for the input configuration only the L1Cfg parameter is used. Other parameters will be described in next sections.

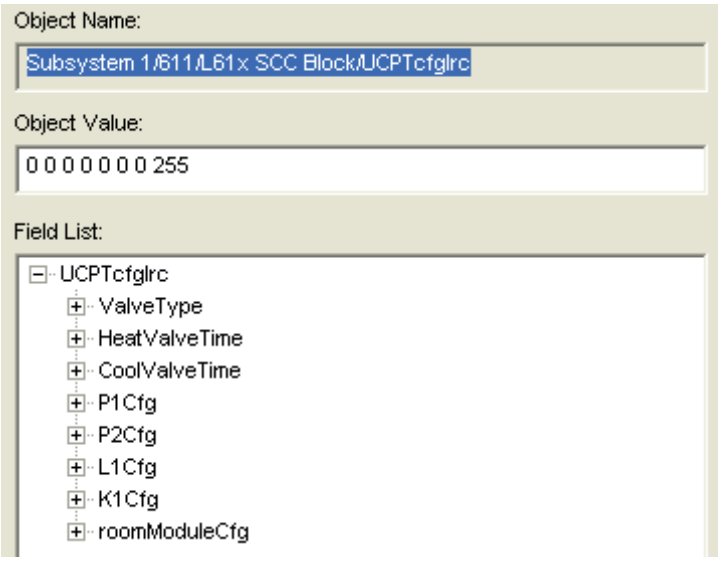


.L1Cfg	Configuration for the R input function.	
	0	Used with an analogue room operation unit (Occupancy state output)
	1	Presence detector (closed = presence)

4.2.3 Analogue Outputs

The next table described outputs available on the PCD7.L611. Each output can be used dependently of your configuration (application configuration and valve type).

Output	Pin	230V	0-10V	Switch	Internal description
K	K1-K2			**X**	Electric heater relay K
Y3	Y3		**X**		0 – 10V output associated to Reg1
Y4	Y4		**X**		0 – 10V output associated to Reg2
Y1	Y1	**X**			Triac on Y1 associated to Reg1 or 3 points valve
Y2	Y2	**X**			Triac on Y2 associated to Reg2 or 3 points valve
V1	V1	**X**			Fan speed V1
V2	V2	**X**			Fan speed V2
V3	V3	**X**			Fan speed V3

nciCfgIrc	Allows configuration of cooling and heating outputs type.	
		
	.ValveType	Configuration of the valves type For a 3 points valve, .HeatValveTime needs to be set !
		0 PWM valve
		1 3 points valve
		2 0-10V valve
	.HeatValveTime	Opening time for a 3 points valve. This variable is in sec and is bordered from 10s to 255s.
	.CoolValveTime	Not used.

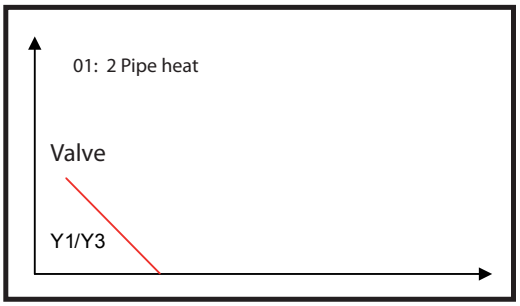
About valves usage:

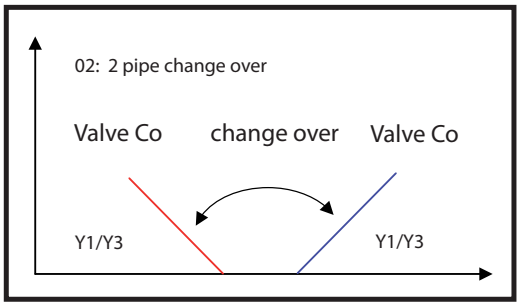
- When switching from triac Y1 active to triac Y2 active, a downtime of 1 second is respected.
- When total closing or opening requests are done on 3 points valve (command to 0% or 100%), the valve cycle time is respected before to consider another command.

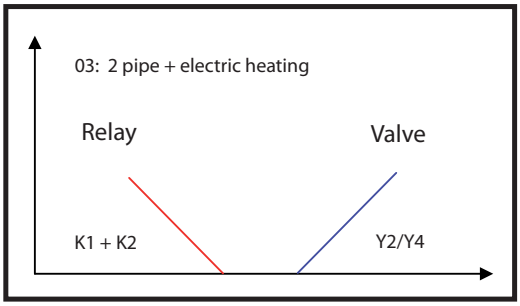
4.3 Application configuration

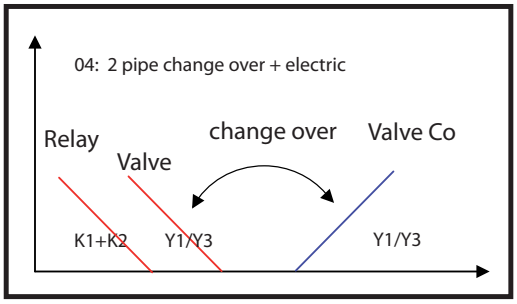
This chapter describe the configuration and the functioning of the HVAC regulator.

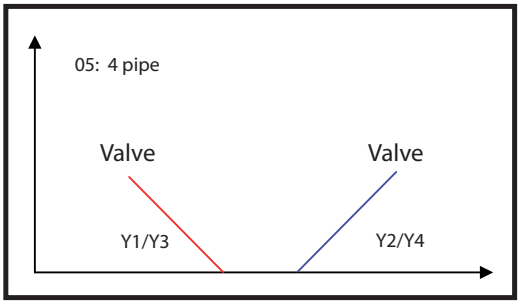
To be adaptable to much kind of installations, the application type needs to be setup in first. This is defined in the configuration variable nciCfgFcc.type. Supported application types are:

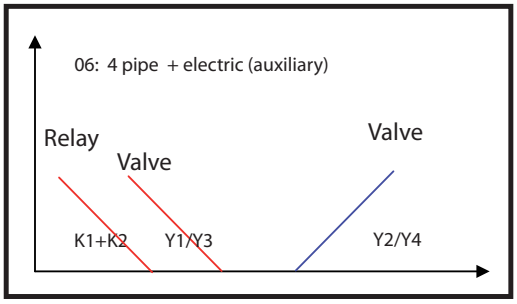


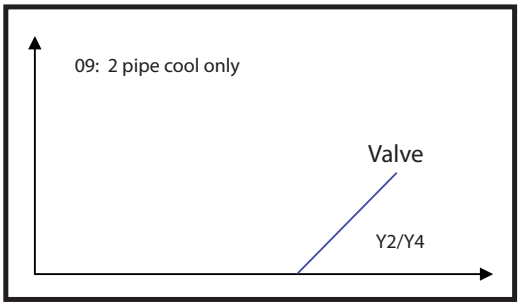












For change over state, valve is in cool mode when **nviChgOver.state = 1** and in heat mode when **nviChgOver.state = 0**.

In next parts of this chapter, basic functions are described. These allow a quick setup of the controller by focusing only on them which are necessarily used for integration. For each functions, variable for configuration are described in first, followed by input and output variables for using this one. Inputs and outputs configuration is mandatory before to start the regulator configuration (chapter “4.1. Inputs / Outputs configuration”).

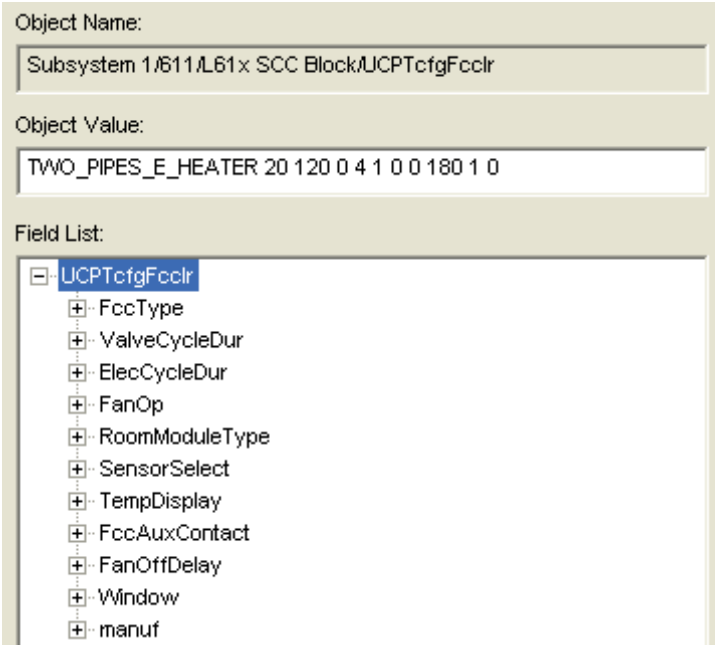
However, it is strictly recommended to configure all options and functions listed in this documentation to be sure to obtain the operation which you want.

4.3.1 Regulator configuration

For the configuration of the regulator part, main variable are listed below. These are used to configure which kind of installation the regulator has to managed, with basic parameters like set points, time cycle of valves or parameters for the PI regulator as the proportional band and the integral time.

In the following description, only variables for HVAC regulator configuration are described.

nciCfgFcc	Used to define the installation type and as the same time the duration of the post ventilation. Other parameters are used for the room operation unit configuration.

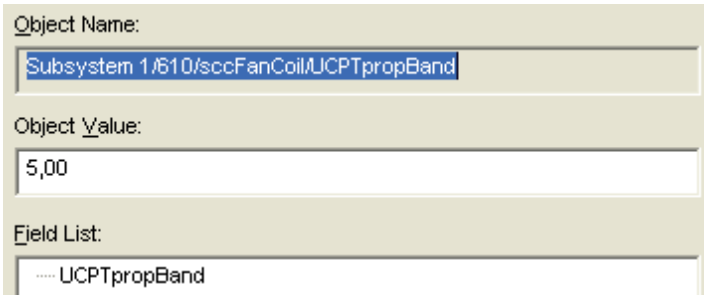


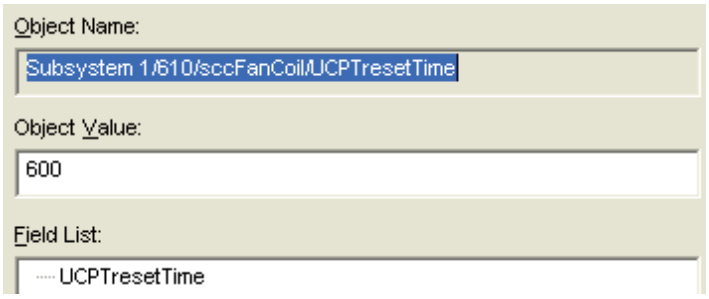
.FccType	To specify the installation type managed by the PCD7.L610.

Type	Description	Ch-Over on Y1/Y3	Heat valve on Y1/Y3	Cool valve on Y2/Y4	Electric heater relay
01	2 pipe heat				
02	2 pipe change over				
03	2 pipe + electric heating				
04	2 pipe Change over + electric heating	primary			secon-dery
05	four pipe				
06	4 pipe + electric heating (aux heat)		primary		secon-dery
09	2 pipe Cooling				

.ValveCycleDur	Time used as the valve cycle time. It is applied to valves configured in PWM (See Chapter “4.1.3. Analogue outputs”).In the case of a 3 points valve, this time is not considered (see nciCfgIrc). This value is in sec and is bordered from 20s to 250s.
.ElecCycleDur	Time used for the PWM cycle of the electric heater. This value is in sec and is bordered from 100s to 250s.
.FanOffDelay	Duration of the post ventilation function. It is used before to stop the fan, as well on a regulation order than on a user forcing. This value is in sec and is bordered from 0s to 255s.

The usage of Y1 or Y3 and Y2 or Y4 depend of the configuration of the valve type (chapter “4.1.2. Analogue outputs”).

ncPropBand	Value used in the PI regulator for the proportional part. This value is in °C and is bordered from 2°C to 20°C.
	

nciIntTime	Value used in the PI regulator for the integral part. To disable the integral part, setup this parameter to 0s. Values below 20s will be considered as 0s and disable the integral part. This value is in sec and is bordered from 20s to 6553s.
	

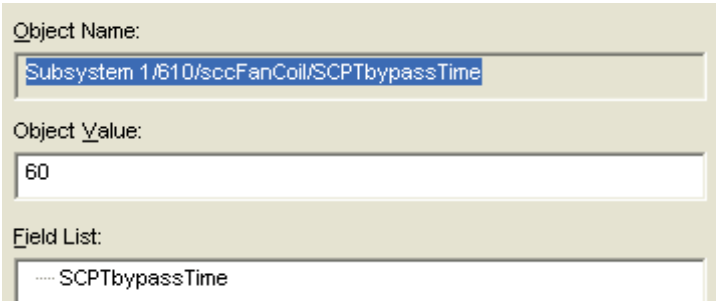
4.3.2 Occupancy mode management

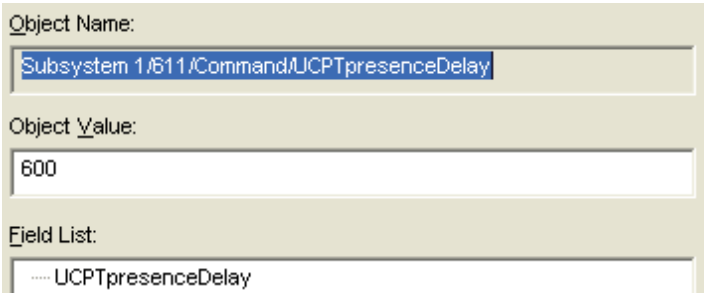
The occupancy mode results from the synthesis of 2 information:

Base mode	Occupancy mode is sent by the BMS or a time schedule. This value has to be written in the **nviOccManCmd**.
Forcing mode	To use the forcing mode, you can write the occupation state by the network to the variable **nviOccSensor** or with a room operation unit (which write in the **nviOccSensor** too). The forcing value is copied out to the **nvoOccManCmd**, and considered during the nciBypassTime. After that, the command is reset to the **nviOccManCmd** value. It is possible to copy the value of the presence detection in the **nviOccSensor** by setting the nciLumCmdPres.Control to 1.

The effective occupation state is given by the variable **nvoEffectOccup** after the computation of these 2 modes. Details about this computation are given by the next table; any other command will be processed according to the rules of the occupied mode.

Base mode	Forcing mode	Effective occupation
nviOccManCmd	**nviOverrideOcc** or local control device (**nvoOccManCmd**)	**nvoEffectOccup**
OC_NUL	OC_NUL	OC_OCCUPIED
OC_NUL	OC_NUL	OC_OCCUPIED
OC_NUL	OC_NUL	OC_UNOCCUPIED
OC_NUL	OC_OCCUPIED	OC_OCCUPIED
OC_NUL	OC_UNOCCUPIED	OC_OCCUPIED
OC_NUL	OC_UNOCCUPIED	OC_UNOCCUPIED
OC_OCCUPIED	OC_NUL	OC_OCCUPIED
OC_OCCUPIED	OC_OCCUPIED	OC_OCCUPIED
OC_OCCUPIED	OC_UNOCCUPIED	OC_UNOCCUPIED
OC_UNOCCUPIED	OC_OCCUPIED	OC_OCCUPIED
OC_UNOCCUPIED	No effect	OC_OCCUPIED
OC_UNOCCUPIED	OC_UNOCCUPIED or OC_NUL	OC_UNOCCUPIED
OC_STANDBY	OC_OCCUPIED	OC_OCCUPIED
OC_STANDBY	No effect	OC_OCCUPIED
OC_STANDBY	OC_UNOCCUPIED or OC_NUL	OC_STANDBY

nciBypassTime	Value of the time to maintain the forcing value passed by the room operation unit or written on **nviOverrideOcc**. The value 0 is interpreted as an unlimited forcing. This value is in minute and is bordered from 0min to 255min.
	

nciPresenceDelay	Time during which the room is considered as occupied after a presence detection. After each detection, the timer is restarted. The value 0 is interpreted as 10 seconds. This variable is in sec and is bordered from 0s to 6553s.
	

nviOccManCmd	The **nviOccManCmd** variable defines the operating mode sent by the BMS. Each time a new value of the **nviOccManCmd** variable is received, the ventilation is forced in automatic mode.

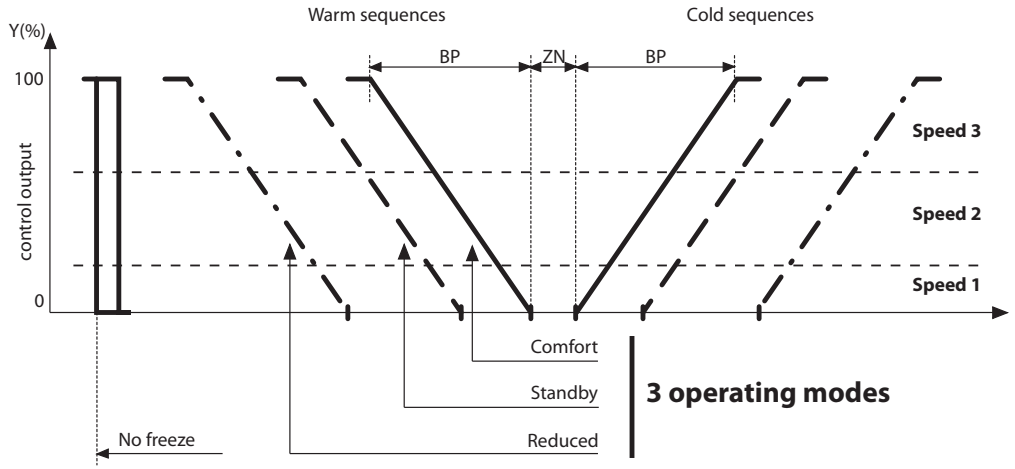
nviOccSensor	The **nviOccSensor** variable is used to force the occupation state by the network. This action can also be done with a room operation unit which automatically update this variable with orders from the user.

nviEffectOccup	Effective occupancy state of the controller used for the regulation. At power-up **nvoEffectOccup** is set to OC_OCCUPIED, due to states of **nviOccManCmd** and **nviOccSensor**.

nvoPresence	This variable is used to provide on the LON network the presence state of the controller and for the light and sunblind management(see chapter “4.4.3. Light & sunblind commands”). The detection sets **nvoPresence** to OC_OCCUPIED during the time configured in the **nciPresenceDelay**. Then **nvoPresence** is reset to OC_UNOCCUPIED. At power-up **nvoPresence** is set to OC_NUL. A debouncing time of 5 seconds is respected after a detection to consider a new one.

4.3.3 Set point adjustment

The evolution of the set point depends principally of the effective occupancy of the room. You can view on the next figure set points for heating and cooling in each occupancy state.



We can identify 3 different cases for the computation of the set point: “Comfort”, “Standby” and “Reduced”.

The effective occupancy state, **nvoEffectOccup**, is used to switch between three main operating modes.

Occupied (**nvoEffectOccup** = OC_OCCUPIED): Comfort operating mode

Stand by (**nvoEffectOccup** = OC_STANDBY): Stand by operating mode

Unoccupied (**nvoEffectOccup** = OC_UNOCCUPIED): Reduced operating mode

If a valid set point is specified for the **nviSetpoint**, it is not directly take in account as the new set point value. It is used to change the central set point value to the **nviSetpoint** value for the occupied mode. An offset value is calculated with the following expression and considered only if the occupation state is set to occupied or standby. This offset is used to change the central set point value to the **nviSetpoint** value for the occupied mode

$$\text{BMSOffset} = \textbf{nviSetpoint} - \frac{\text{nciSetpoints.occupied}_{cool} + \text{nciSetpoints.occupied}_{heat}}{2}$$

Occupied (nvoEffectOccup = OC_OCCUPIED) or Bypass (nvoEffectOccup = OC_BYPASS) mode

- Warm set point = nciSetpoints.occupied_heat + **nvoSetptOffset** + BMSOffset
- Cold set point = nciSetpoints.occupied_cool + **nvoSetptOffset** + BMSOffset

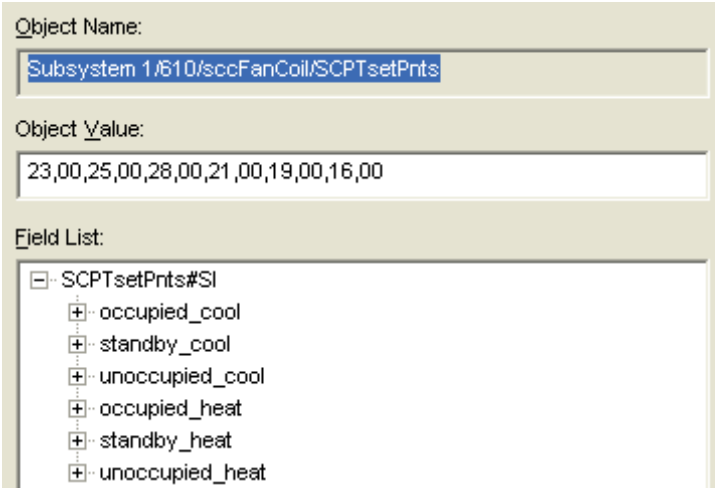
Santdby (nvoEffectOccup = OC_STANDBY) mode

- Warm set point = nciSetpoints.standby_heat + **nvoSetptOffset** + BMSOffset
- Cold set point = nciSetpoints.standby_cool + **nvoSetptOffset** + BMSOffset

Unoccupied (nvoEffectOccup = OC_UNOCCUPIED) mode

- Warm set point = nciSetpoints.unoccupied_heat
- Cold set point = nciSetpoints.unoccupied_cool

For each occupation mode, the regulation dead zone is fixed between these 2 set points.

nciSetpoints	Values for the computation of the effective set point. All of these values are in °C and are bordered from 10°C to 35°C.
	

nviSetpoint	Set the central set point (middle of dead zone) in occupied mode. The regulator updates the heat and cool set point values with the BMSOffset compute in occupied mode and standby mode too. This value is in °C and is bordered from 5°C to 40°C.

nviSetptOffset	Offset value for the set point. It is considered only if the occupation state is set to Occupied or Standby. If this variable is bound and the controller is configured with an analogue room operation unit, Offset set point orders from the room operation unit are not considered. This value is in °C and is bordered from -10°C to 10°C.

nvoEffectSetpt	Value used by the regulator as effective set point. This value is in °C.

nvoSetptOffset	Actual offset considered for the computation of the effective set point. This value can be set by the user with the room operation unit or by the BMS with the **nviSetptOffset**. Only the last write of one of these two actions is taken in account. This value is in °C and is bordered from -10°C to 10°C.

4.3.4 Temperature

The temperature measurement may come from various devices:

- A temperature probe directly connected to the controller (on screw terminals).
- A remote controller or a room operation device directly connected to the controller via the RJ9 link.
- Other devices on the network.

The controller manages the following priorities:

0 Network variable if the variable **nviSpaceTemp** is valid (-10°C< Value <65°C).

1 Temperature sensor configured by default for the controller in the **nciCfgSrc. SensorSelect** (see chapter 4.1.1 Room operation unit).

2 If in addition of the default temperature sensor (RJ9 if **nciCfgSrc.SensorSelect** = 0 or analogue probe if **nciCfgSrc.SensorSelect** = 1) another probe (from type of the one which is **NOT** configured) is connected, its value can be used. It is considered with the last priority, only if invalid temperature is present on both temperature inputs with priority 0 and 1.

For an analogue sensor connected on screw terminals, the measure is filtered to be considered only if its value is comprise from 0°C to 90°C.

If the sensor temperature used is on the RJ9 link, its value will be sent periodically to the controller (depending on its variation). If this value is not received for more than 4 hours (250 minutes exactly), and the controller does not have another valid temperature, the **nvoSpaceTemp** is set to 327.67°C (invalid temperature) and the regulation is stopped.

If no measurement temperature is valid, the **nvoUnitStatus.in_alarm** variable is set to 1.

nviSpaceTemp	Variable used to receive a temperature from the BMS or from another device from the network. This value is in °C and is bordered from -10°C to 65°C.

nvoSpaceTemp	Temperature used by the controller for the regulation. It can be equal to the **nviSpaceTemp** or take its value for its default sensor more the value of the offset sensor (nciOffsetTemp). This value is in °C and is bordered from -10°C to 65°C.

4.3.5 Regulation in use

The computation of the control loop and the update of regulation variable are done every 10 seconds. However, to achieve fast response time for critical actions, the control loop execution is forced in the following cases:

- Modification of the fan speed (**nviFanSpeedCmd** or room operation device).
- Modification of the contact states (**nvoWindow** or **nviEnergyHoldOff**).

When the regulator is in used, it is possible to check regulation status and to act on them. For this, you have to use the following variables.

nviApplicMode	To act on the application mode. Following modes are supported by the device.
	HVAC_NUL (-1): not take in consideration. HVAC_AUTO (0): the operating mode is determined by the controller. HVAC_HEAT (1): warm mode forcing. HVAC_COOL (3): cold mode forcing. HVAC_OFF (6): controller stop, frost guard mode still active. HVAC_TEST (7): test mode, used to force state outputs. HVAC_EMERG_HEAT (8): warm emergency, used by the frost guard mode All others: warm mode forcing.

nviEnergyHoldOff	Used to enable or stop the control loop (see chapter 4.3.4).

nvoEnergyHoldOff	State of the control loop (see chapter 4.3.4).

nvoHeatCool	State of the effective application mode of the controller.

nvoOutputPrimary	State of output used for cooling (refer to table 4).

nvoHeatPrimary	State of output used for heating (refer to table 4).

nvoUnitStatus	State of the control loop.

4.4 Functions

All modifications on configuration variables are not consider immediately or on the next execution of the control process loop. It is highly recommended to restart the device after the complete configuration to be sure to active all new configurations. This can be done by unplug and plug again the power supply connector or by the network.

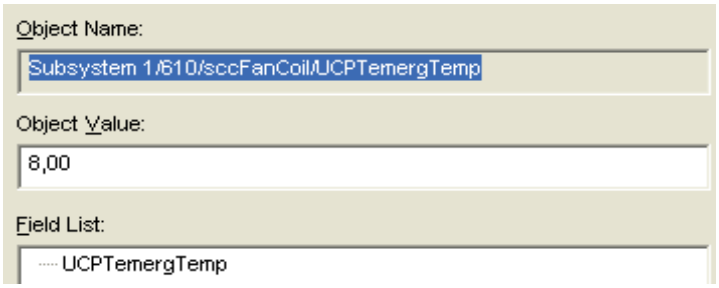
4.4.1 Frost-guard mode

This mode has the higher priority on any other mode or function and is always active.

If room temperature < Antifreeze limit (**nvoSpaceTemp** < **ncEmergTemp**), then the fan speed is set to its maximal value, heat valve and electric battery are forced to 100%.

When antifreeze actions are enabled, **nvoHeatCool** = HVAC_EMERG_HEAT.

This application mode is active as long as the room temperature is not higher than the antifreeze temperature more 1°C (hysteresis threshold).

ncEmergTemp	Variable to define the threshold for engaging the frost guard mode. This value is in °C and is bordered from 0°C to 20°C.
	

4.4.2 Ventilation output control

Ventilation can be used in automatic mode or in forced mode.

For automatic mode, the fan speed is managed by the regulator according to the use of heating and cooling outputs.

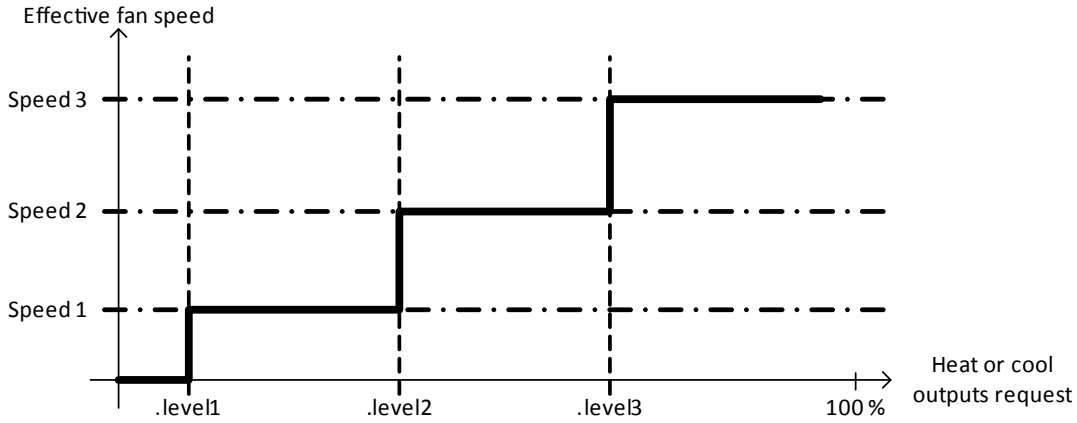
For forced mode, orders can be sent from the room operation unit or by the network, using the **nviFanSpeedCmd**. Orders sent can be viewed in the **nvoFanSpeedCmd** whereas the effective state of the fan is placed in the **nvoFanSpeed**.

Before to stop the ventilation, post ventilation is necessarily respected. During this time, the fan keeps in speed 1 during the time configured in the **nciCfgSrc. FanOffDelay**. This security can't be disabled but you can reduce its time to the minimal value, 10s.

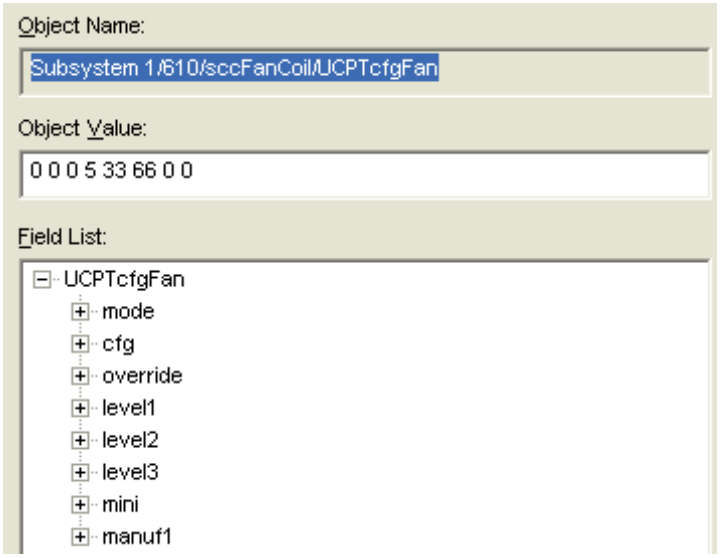
If the ventilation is forced to stop with the room operation unit, the regulation is stopped at the same time, unless it is in frost guard mode. The post ventilation time is still kept before to force the ventilation to stop.

When the regulation is in the dead zone, ventilation is stopped in automatic mode. If you want to force the speed fan to 1 in this zone used the **nciCfgFcc.FanOp**. This forcing is not done when **nvoEnergyHoldOff.state**=1 (when an opening window is detected for example). It is also possible to force ventilation always to stop or according to the application mode (heating and cooling) with the variable **nciCfgFan.FanOp** to.

On the PCD7.L610, only 3 speed fans are supported. In this mode, the fan shifts between its 3 gears depending on the regulation request in **nvoUnitStatus** (see chapter "4.2.5. Regulation in use"). Thresholds for engaging each speed are configurable with the **nciCfgFan.levelX**. The ventilation is switched off for 1s between each speed.

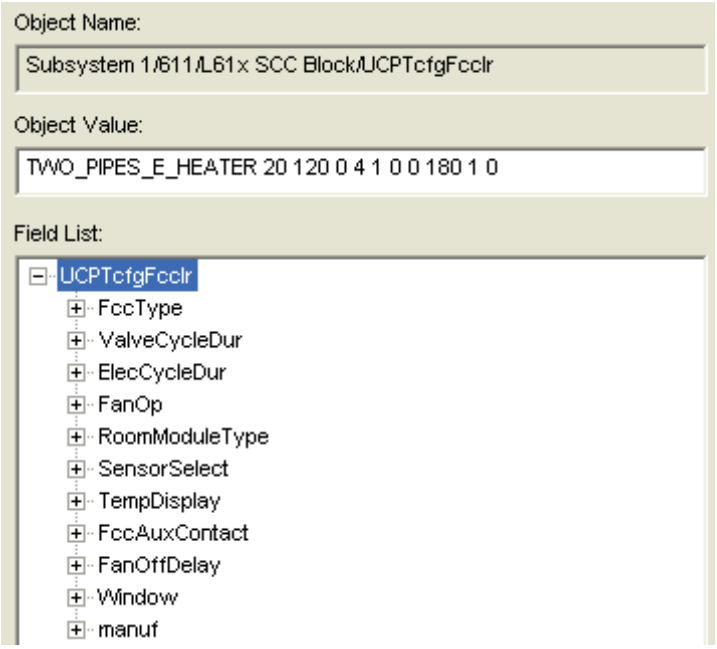


nciCfgFan	Allow to configure ventilation thresholds for changing fan speed.



.mode	Not used
.cfg	Not used
.override	Not used
.level1	Threshold on regulation demand to switch the fan in speed 1 (considered in automatic mode only). This value is in % and is bordered from 0% to 100%.
.level2	Threshold on regulation demand to switch the fan in speed 2 (considered in automatic mode only). This value is in % and is bordered from 0% to 100%.
.level3	Threshold on regulation demand to switch the fan in speed 3 (considered in automatic mode only). This value is in % and is bordered from 0% to 100%.
.mini	Not used.
.manuf1	Not used.

nciCfgFcc	Allow to configure ventilation forcing and post ventilation time.



.FanOp	Forcing mode of the ventilation.	
	0	No forcing
	1	Speed 1 in dead zone if occupied or standby
	2	Speed 1 in dead zone
	3	no ventilation in warm mode
	4	no ventilation in cold mode
	5	no ventilation, regardless of the regulation mode
.FanOffDelay	Duration of the post ventilation function. It is used before to stop the fan, as well on a regulation order than on a user forcing. This value is in sec and is bordered from 0s to 255s.	

nviFanSpeedCmd	Used to force the fan speed.

nvoFanSpeed	Display the effective fan speed.

nvoFanSpeedCmd	Display the fan speed forced by the room operation unit or by the **nviFanSpeedCmd**.

nviFanSpeedCmd, **nvoFanSpeed** and **nvoFanSpeedCmd** are based on the SNVT_switch format which is composed of 2 fields, “state” and “value”. These variables use the SNVT_switch in concordance with the next table.

State	Value	Description
-1	0	Auto
0	0	Stop
1	33	Speed 1
1	66	Speed 2
1	100	Speed 3

4.4.3 Change Over

Depending on the application configuration, one valve can be used in change over mode (see Reg 1 in chapter “4.2. Application configuration”). In this case, the valve can supply cold or warm depending on the change over state.

To manage the change over state 2 possibilities are available, the first is the network variable **nviChgOver** and the second is the input E2 in change over configuration (see chapter “4.1.2. Analogue inputs“). This state is displayed by the **nvoChgOver**.

nviChangeOver	To forced the state of the change over.

nviChgOver and **nvoChgOver** are based on the SNVT_switch format which is composed of 2 fields, “state” and “value”. These variables use the SNVT_switch in concordance with the next table.

State	Value	Description
0	0	Warm mode
1	100	Cold mode

4.4.4 Window or door contact processing

The room controller embeds by default an input configured for window or door contact (input E1). It is used to detect an open window or door regardless of the contact polarity (managed with the **nciCfgFcc.Window**). In this case, the regulation is stopped (valve closed, fan and electric battery stopped) but frost guard mode is still active.

The detection of the open window can be done by two ways:

- Contacts plugged on the E1 input (see chapter “4.1.2. Analogue inputs”). In this case, the state of the contact is displayed by the **nvoWindow**.
- By the Lon network with the **nviWindowLoop** variable.

When a window opening is detected, the **nviEnergyHoldOff** is updated either with the **nvoWindow** or with the **nviLoopWind** on which is the latest updated. Usage of both ways at the same time is not advised, unless it is for a master/slave configuration (see chapter 4.3.15 Master / Slave).

The **nviEnergyHoldOff** variable and the window contact (**nvoWindow**) are used to determine if a window is opened.

In this mode, the regulator doesn’t allow fan speed forcing, doesn’t consider the room operation unit commands and stops (if configured), the small speed fan forcing into the dead zone.

When using a bidirectional room operation unit with a LCD display, an alarm is displayed on the screen.

The window contact input is filtered (debouncing).

nviEnergyHoldOff	Energy saving command. This command can be used with the window contact information.

nviLoopWind	Window contact information for looping when several controllers are present in the same room (see chapter “4.3.15. Master/Slave”).

nvoEnergyHoldOff	Result for computation of the opening window process control.

nvoWindow	Actual window contact state of the controller.

nviWindowLoop and **nvoWindow** are based on the SNVT_switch format which is composed of 2 fields, “state” and “value”. These variables use the SNVT_switch in concordance with the next table. These values are used for **nviEnergyHoldOff** and **nvoEnergyHoldOff** too.

State	Value	Description
0	0	Window closed, normal operation
1	100	Window open, control loop disabled

4.4.5 Dew point

In cooling mode, dew may form on the cooling register. To prevent this, a dew sensor can be used with the regulator. When condensation is detected, the cold output of the regulator is forced to 0 but the control loop is still active. Computations of outputs are still done by the PI regulator; the ventilation follows the control process signal or the ventilation forcing parameters.

To activate this function, the auxiliary contact needs to be configure as a dew point detector with the **nciCfgFcc.FccAuxContact**=3. This functionality managed the input as normally open (NO).

nvoAuxContact	Displayed the input state of the sensor plugged on analogue contact.

nvoAuxContact is based on the SNVT_switch format which is composed of 2 fields, “state” and “value”. This variable use the SNVT_switch in concordance with the next table.

State	Value	Description
0	0	Dew detection active
1	100	Normal operation

4.4.6 Actions of contacts on the process control loop

This table is a simply sum up of chapters “4.3.3. Change over” to “4.3.7. Flow control”.

Window	nciCfgFcc.window	nvoWindow	Effect
contact “open”	0	{0 0}	Process control loop is active
contact “open”	1	{1 100}	Process control loop is stopped
contact “closed”	0	{1 100}	Process control loop is stopped
contact “closed”	1	{0 0}	Process control loop is active

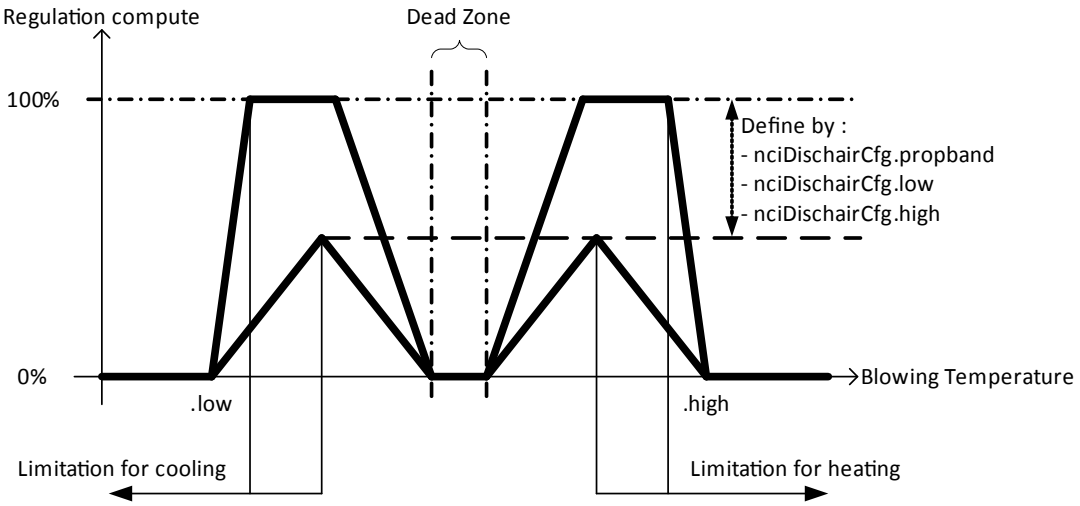
Dew	nciCfgFcc.FccAux Contact	nvoAuxSensor	
contact “open”	3	{0 0}	Warm process control only – cold mode is stopped
contact “closed”	3	{1 100}	No effect

Change Over	nciFunctionCfg.chgover	nvoChgOver	
contact “open”	2	{100 1}	Cold mode
contact “closed”	2	{0-0}	Warm mode

4.4.7 Blowing temperature limitation

This function is available only if the auxiliary contact is configured as a blow temperature sensor with the **nciCfgFcc.FccAuxContact**=5.

This function can be used to limit the temperature of the air blown by the device during the regulation. It permits to define two thresholds, one for warm air and one for the cold air. Each time the blowing temperature come up to these limits, valves or the electric heater are limited then stop when limits are reached. Limitations can be described with the next figure.



The blow limitation can only be used if a valid temperature is measured by the dedicated sensor, **nvoAuxSensor** different of 327.67°C (see chapter 4.1.2. Analogue inputs"). In this case, the limitation can be applied on heating, cooling or both application modes with the nciDischair.type.

- For cooling limitation: To limit the cold air temperature, the low limit needs to be used (**nciDischair.low**). The limitation will passed by 3 states during the decrease of the dish air temperature.

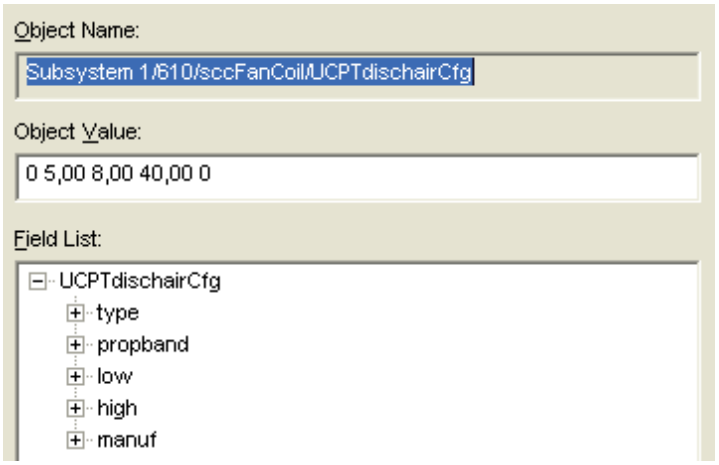
  - **nvoAuxSensor > nciDischair.low + nciDischair.propband**:
    The regulation works normally, no limitation applied.

  - **nvoAuxSensor < nciDischair.low + nciDischair.propband**:
    Limitation of the cold output proportionally to the difference with the low limit.

  - **nvoAuxSensor < nciDischair.low**:
    Cold output forced to 0%.

- For heating limitation: To limit the warm air temperature, the high limit needs to be used (**nciDischair.high**).
  - **nvoAuxSensor < nciDischair.high - nciDischair.propband:**
    The regulation works normally, no limitation applied.
  - **nvoAuxSensor > nciDischair.high - nciDischair.propband:**
    Limitation of the warm output proportionally to the difference with the high limit
  - **nvoAuxSensor > nciDischair.high:**
    Warm output forced to 0%

nciDischair	Used to enable the blow limitation function and define level limitation used by this one.



.type	Define which limits are enabled for the blow limitation.	
	0	No limitation.
	1	Low limitation is active.
	2	High limitation is active.
	3	Both limitations are active.
.propband	Proportional band used to limit outputs before to force them to 0.	
.low	Value of the low limit. This value is in °C and is bordered from 0°C to 99°C.	
.high	Value of the high limit. This value is in °C and is bordered from 0°C to 99°C.	
.manuf1	Not used.	

ncAuxSensor	Temperature measure by the blowing air temperature sensor (only if the nciCfgFcc.FccAuxContact=5). This value is in °C and is bordered from 0°C to 99°C.

4.4.8 Control of the electric heater ventilation

The usage of the electric heater is limited, when its demand is under 85% it is always used as 100%.

If the manual command of the fan speed leads to a stop of the fan, the electric battery request is forced to zero. Likewise, if the stop of the fan is caused by the forcing from the **nciCfgFcc.FanOp**, the electric battery is stop.

The operation time for the electric heater is displayed by the **nvoHeaterRunTime**. This value is stored in the EEPROM memory of the device every 12 hours. If a reset occurred, this value is reloaded from the EEPROM memory. To reset it, used the **nviRequest** with the value **nviRequest** = 0,RQ_CLEAR_RESET.

nvoHeaterRunTime	Electric heater operation time. This value is in hour and is bordered from 0 hour to 65535 hours.

4.4.9 Forced variable propagation and receive heartbeat

To control the network load, it is possible to configure a heart bit value for the propagation of some variables. With this function, variables can be propagated even if their values haven't change.

The **nciSndHrtBt** defines at which time variables are sent. This heartbeat is applied to:

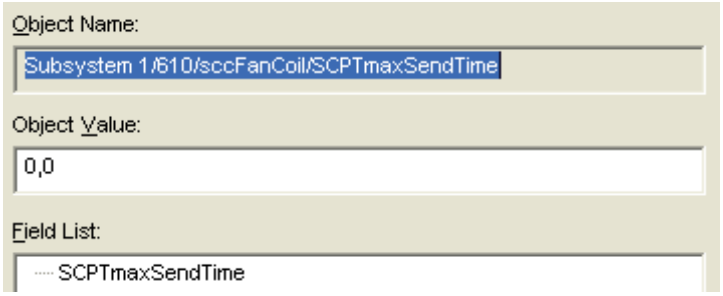
- **nvoAuxContact**
- **nvoCoolPrimary**
- **nvoEffectOccup**
- **nvoFanSpeed**
- **nvoHeatCool**
- **nvoHeatPrimary**
- **nvoSpaceTemp**
- **nvoWindow**

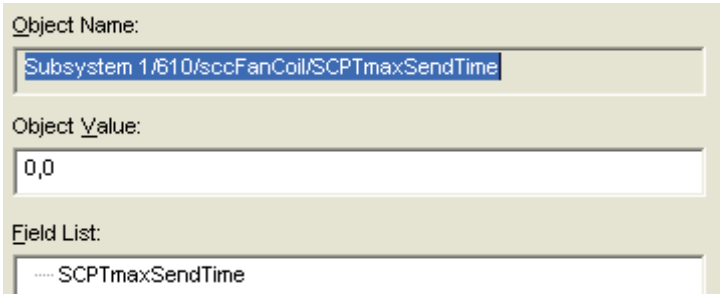
For the receive heartbeat, this security is applied only on the following variable and only if they are bound. If the variable is not received at the end of the **nciRcvHrtBt**, its value is set to invalid. This heartbeat is applied to:

- **nviApplicMode**, set to invalid value "HVAC_AUTO"
- **nviSpaceTemp**, set to invalid value "327,67°C" (update at the same time **nvoSpaceTemp**).

Both functions can be disabled with the value 0s.

These functions are mainly used in Master / Slave mode
(see chapter "4.3.11 Master / Slave").

nciRcvHrtBt	Heartbeat for update on associated variable variables. This value is in sec and is bordered from 0s to 6553s.
	

nciSndHrtBt	Heartbeat value for propagation of associated variables. This value is in sec and is bordered from 0s to 6553s.
	

4

4.4.10 Electric heater limitation / Load shedding

It is possible to limit the electric battery power by using **nviEconEnable**. The power limitation can be used to reduce its consumed power or to stop it.

- if **nviEconEnable.state** = 0, no power limitation.
- if **nviEconEnable.state** = 1, power is limited to **nviEconEnable.value**.

nviEconEnable	Used to manage the load shedding for the electric heater. It is based on the SNVT_switch format which is composed of 2 fields, “state” and “value”. These variables use the SNVT_switch in concordance with the next table.

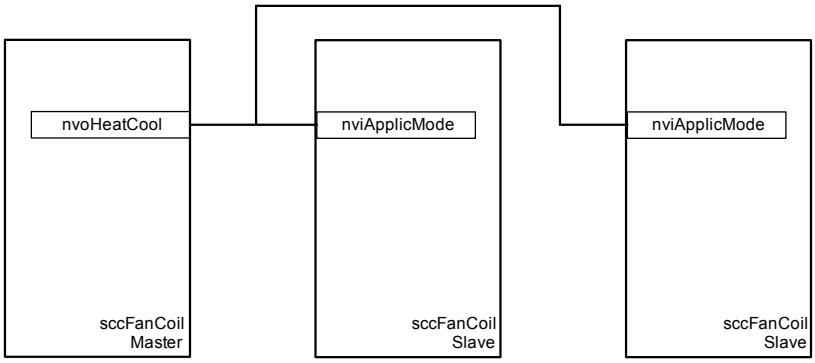
State	Value	Load shedding	nviEconEnable.value format
0	0	No load shedding	/
1	X	Electric heater output limited to X%	Percent – %
1	0	Electric heater stopped	Percent – %

4.4.11 Master / Slave

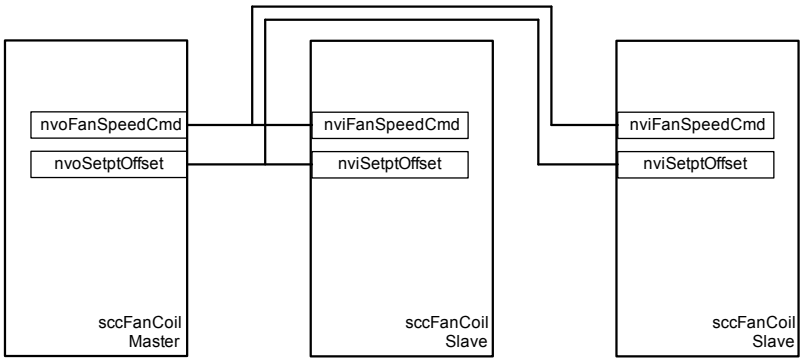
When several controllers are installed in the same room, it is necessary to have a consistency in the operation of these controllers. At this end, a controller will be defined as the « master » and this master will send at least the operating mode to the other controllers defined as the “slaves”:
nvoHeatCool will be sent to the slaves to update **nviApplicMode**.

The other bindings will depend on the user control devices which are used (one or several room devices or infrared or radio remote controllers in the same room).

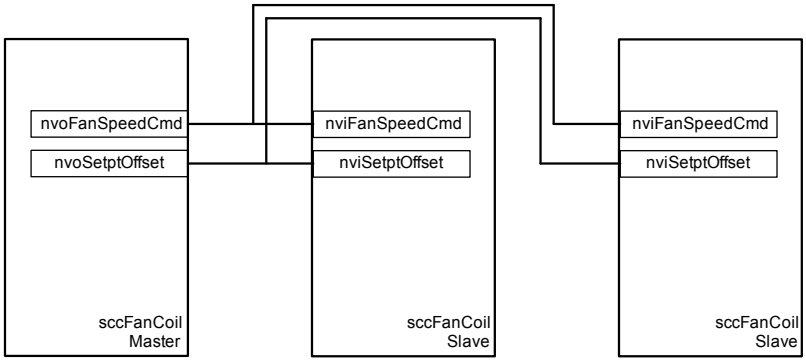
Application mode master/slave links:



Fan speed command and set point offset master/slave links:



Window master/slave links:

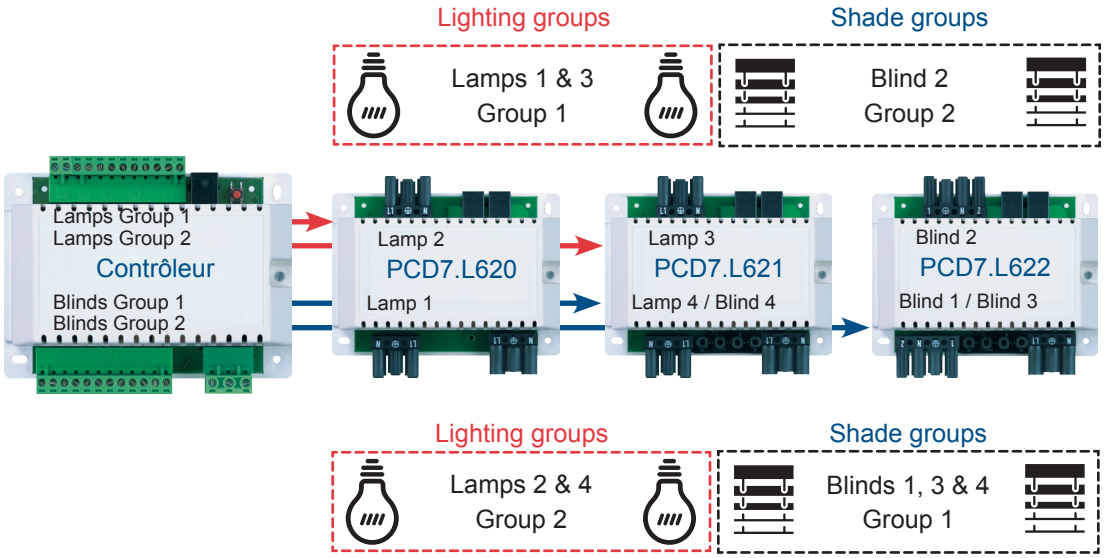


4.5 Light & Sunblind management

4.5.1 Application

The PCD7.L611 can be used with add-on devices to drive light and sunblind. It is compatible with devices:

- PCD7.L620: Add-on 2 “lights” ON/OFF outputs
- PCD7.L621: Add-on 2 “lights” ON/OFF outputs, 1 “sunblind” 230 VAC output
- PCD7.L622: Add-on 3 “sunblind” 230 VAC outputs
- PCD7.L623: Add-on 2 “sunblind” 24Vdc outputs



The PCD7.L611 integrates 4 light objects and 4 sunblind objects to manage these add-on modules.

Each add-on module output is associated to one object in the PCD7.L611.

For light objects

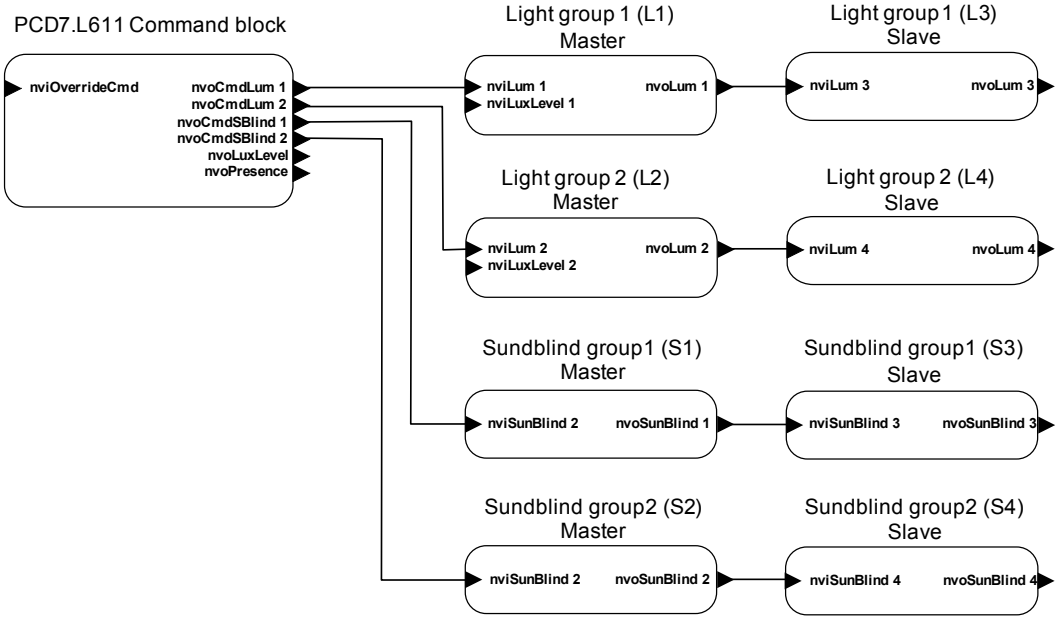
- Lamp1 ➔ L1
- Lamp2 ➔ L2
- Lamp3 ➔ L3
- Lamp4 ➔ L4

For sunblind objects

- SBlind1 ➔ S1
- SBlind2 ➔ S2
- SBlind3 ➔ S3
- SBlind4 ➔ S4

4.5.2 Manufactory settings

The controller is equipped with a default configuration which allows it to work in a predefined way without any network operation. This configuration is still in use unless binding are made on **nvoCmdLumX** or **nvoCmdSBlindX**.



4.5.3 Light & Sunblind commands

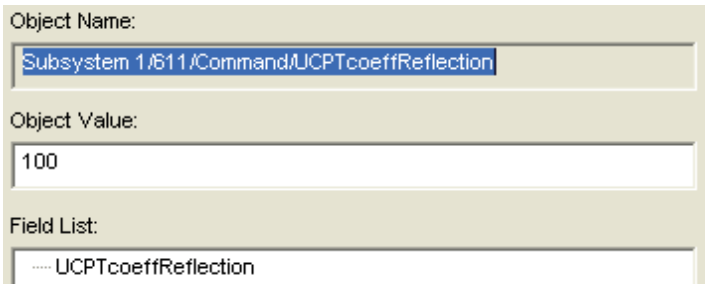
To apply forcing from the room operation unit, the Command object needs to be used like in the default application.

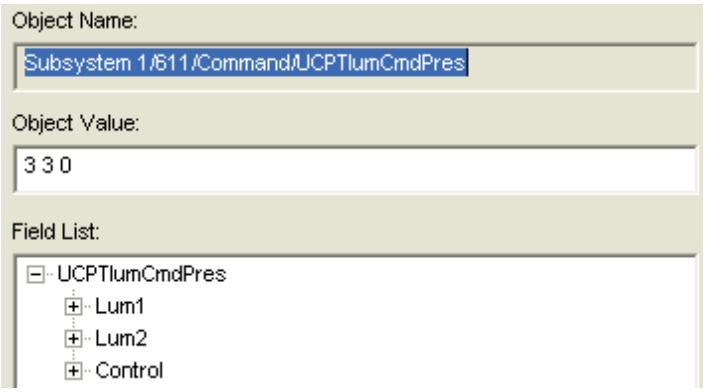
When presence detection occurs, the luminosity level of the room is checked. If this level is under the **nciLumLevelPres**, light forcing from the **nciLumCmdPres** are applied. At the end of the presence detection, light forcing from the **nciLumCmdPres** are applied too. See the description of this variable for more details.

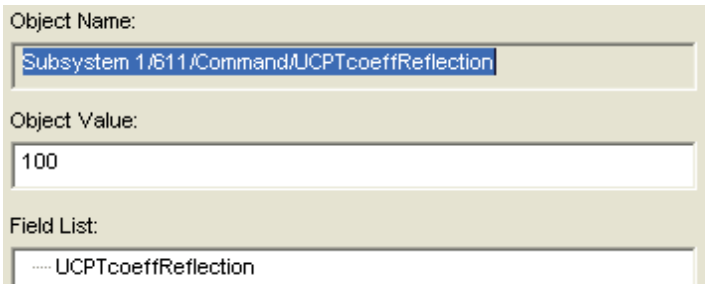
It is possible to copy the value of the presence detection in the **nviOccSensor** by setting the **nciLumCmdPres**.Control to 1.

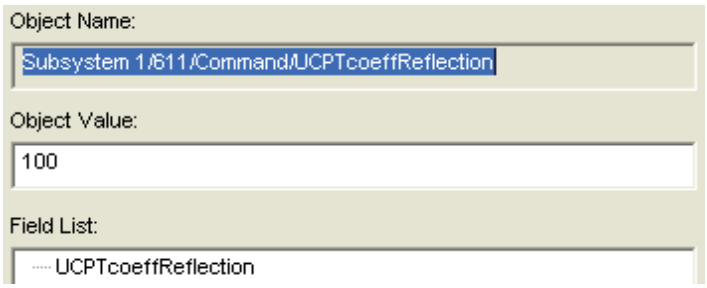
The sensibility of the luminosity sensor needs to be adapted with the **nciCoeffReflection**, depending on its room environment. The result of the effective luminosity level computation is given by the **nvoLuxLevel** and respects the following equation:

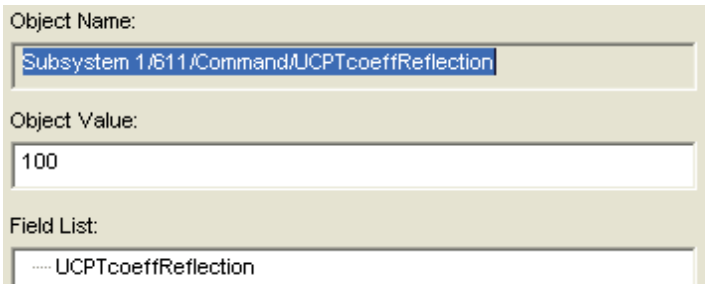
$$Actual\ luminosity = \frac{Measured\ luminosity\ x\ 100}{nciCoeffReflection}$$

nciCoeffReflection	Coefficient to calibrate the luminosity measurement, according to the room environment. This value is in % and is bordered from 1% to 100%.
	

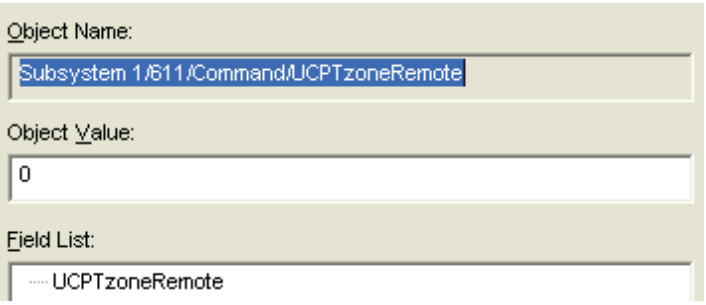
nciLumCmdPres	Use to configure the action of the presence detection on lights and on the HVAC regulation.	
		
.Lum1	effect of presence detection on light 1.	
	0	no effect
	1	light is turned ON if presence
	2	light is turned OFF after timer has elapsed if no presence
	3	light is turned ON and OFF (combination of 1 and 2)
.Lum2	effect of presence detection on light 2.	
	0	no effect
	1	light is turned ON if presence
	2	light is turned OFF after timer has elapsed if no presence
	3	light is turned ON and OFF (combination of 1 and 2)
Control	effect of presence detection on HVAC.	
	0	no effect
	1	occupancy forcing if presence (**nviOccSensor** is updated)

nciLumLevelPres	Define the luminosity level threshold on a presence detection. This unit is in lux and is bordered from 0lux to 1020lux.
	

nciPresenceDelay	Duration of the occupied state for the room after a presence detection. After each detection, the timer is restarted. The value 0 is interpreted as 10 seconds. This variable is in sec and is bordered from 0s to 6553s.
	

nciSBlindTime	Timeout value for the UP or DOWN sunblind movement. This variable is in sec and is bordered from 0s to 6553s.
	

nciZoneRemote	For remote operation unit, a zone address needs to be configured in each one to be sure to act on the correct room controller. This variable allows defining which number for the room operation unit can be takes in account by the PCD7.L610. Its value is bordered from 0 to 30.



0	Universal receiver. Accept each remote control unit, whatever its number.
X (from 1 to 30)	Accept only orders and information from a remote control unit with the same zone address. This setting doesn't serve to configure the zone address into the remote operation unit. It is used only to consider orders with a number which match to this variable. To configure the remote operation unit and its zone address, refer to its own documentation.

nviOverrideCmd	Forcing of the light and sunblind commands. It is based on the SNVT_setting format which is composed of 3 fields, "function" "setting" and "rotation". The "rotation" field is never considered. This variable use the SNVT_setting in concordance with the next table.		
	Function	**Setting**	**Description**
	SET_UP	0	Forcing sun blinds to up*
	SET_DOWN	0	Forcing sun blinds to down*
	SET_ON	0	Forcing lights to ON
	SET_OFF	0	Forcing lights to OFF
	SET_NUL	0	Stop the last sun blinds forcing

*: Local command is disabled until the forcing is applied on the **nviOverrideCmd**.

nvoCmdLumX	Light group X command, from room operation controller or forcing on the **nviOverrideCmd**.

nvoSunBlindX	Sunblind group X command, from room operation controller or forcing on the **nviOverrideCmd**

nvoLuxLevel	Room luminosity level after the computation with the nciCoeffReflection. This value is in lux and is bordered from 0lux to 1020lux.

nvoPresence	This variable is used to provide on the LON network the presence state of the controller and for the light and sunblind management(see chapter “4.4.3. Light & sunblind commands”). The detection sets **nvoPresence** to OC_OCCUPIED during the time configured in the nciPresenceDelay. Then **nvoPresence** is reset to OC_UNOCCUPIED. At power-up **nvoPresence** is set to OC_NUL. A debouncing time of 5 seconds is respected after a detection to consider a new one.

4.5.4 Details on light object

A light object can be used with 4 functions, depending on its lamp type define in the **nciCfgLumX.type**:

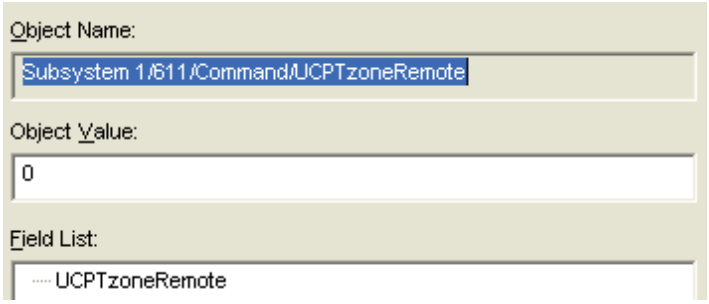
For ON/OFF lamp, **nciCfgLumX.type**=0:
- SET_ON: switch the light ON.
- SET_OFF: switch the light OFF.
- SET_STATE: this function is considered by a ON/OFF light only if its setting filed is set to 0% or 100% and switch the light in the corresponding state (see table 22 “Usage of the **nviLumX**).
- SET_STOP: stop the last function receives by the lamp object and keep the light in its actual state.

For dimming lamp, **nciCfgLumX.type**=1:
- SET_ON: switch the dimming light directly to 0%.
- SET_OFF: switch the dimming light directly to 100%.
- SET_STATE: switch the light in increase or decrease mode, depending on the setting filed (see table 22 “Usage of the **nviLumX**).
- SET_STOP: stop the last function receives by the lamp object and keep the light in its actual state.

For each function, the light object keeps in its last receiving function until the next writing on its **nviLumX**. So for a dimming lamp, the light object continues to increase or decrease its level until it received the “SET_STOP” function.

Slave can be bound on master light by using the **nvoLumX** of the light object. By using this configuration, light master and salve are placed in the same light group and you save “alias” in use on your LON network.

nciCfgLumX	Define the lamp type of the light X.



0	ON/OFF lamp
1	Dimming lamp

nviLumX	Command to use the Light number X. It is based on the SNVT_setting format which is composed of 3 fields, “function” “setting” and “rotation”. The “rotation” field is never considered. This variable use the SNVT_setting in concordance with the next table.

Function	**Setting**	**Description**
SET_ON	0	Switched lights to ON (or to 100% for dimming lamp)
SET_OFF	0	Switched lights to OFF (or to 0% for dimming lamp)
SET_STATE	0	Switched lights in increase mode (or turn off for a ON/OFF lamp)
SET_STATE	100	Switched lights in decrease mode (or turn on for a ON/OFF lamp)
SET_STOP	0	Stop increase or decrease mode on dimming lights

nvoLumX	Display the state of the light X. This variable is used to bind master and slave light.

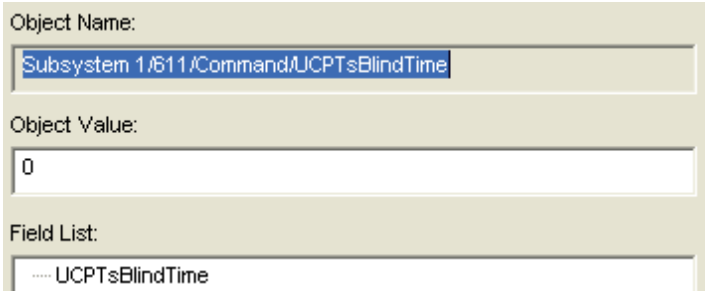
4.5.5 Details on sunblind object

A sunblind object can be used with 3 functions:

- SET_UP: switch the sunblind in the go up way.
- SET_DOWN: switch the sunblind in the go down way.
- SET_STOP: stop the last function receives by the lamp object and keep the light in its actual state.

For each function, the sunblind object keeps in its receiving function until the next writing on its **nviSunBlindX**. So, the sunblind object continues to go up or down until it received the “SET_STOP” function. If it received a “SET_DOWN” order directly after a “SET_UP”, or inversely, the sunblind movement switch directly after a delay of 1 second.

As for the light, slave can be bound on master sunblind by using the **nvoSunBlindX** of the sunblind object.

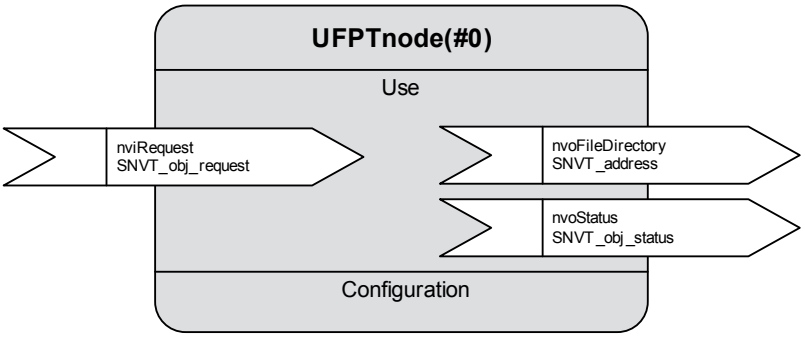
nciSBlindTime	Timeout value for the UP or DOWN movement of the sunblind. This variable is in sec and is bordered from 0s to 6553s.
	

nviSunBlindX	Command to use the sunblind number X. It is based on the SNVT_setting format which is composed of 3 fields, “function” “setting” and “rotation”. The “setting” and the “rotation” field are never considered. This variable use the SNVT_setting in concordance with the next table.

nviSunBlindX	Display the state of the light X. This variable is used to bind master and slave light.

5 Functional blocks and variables

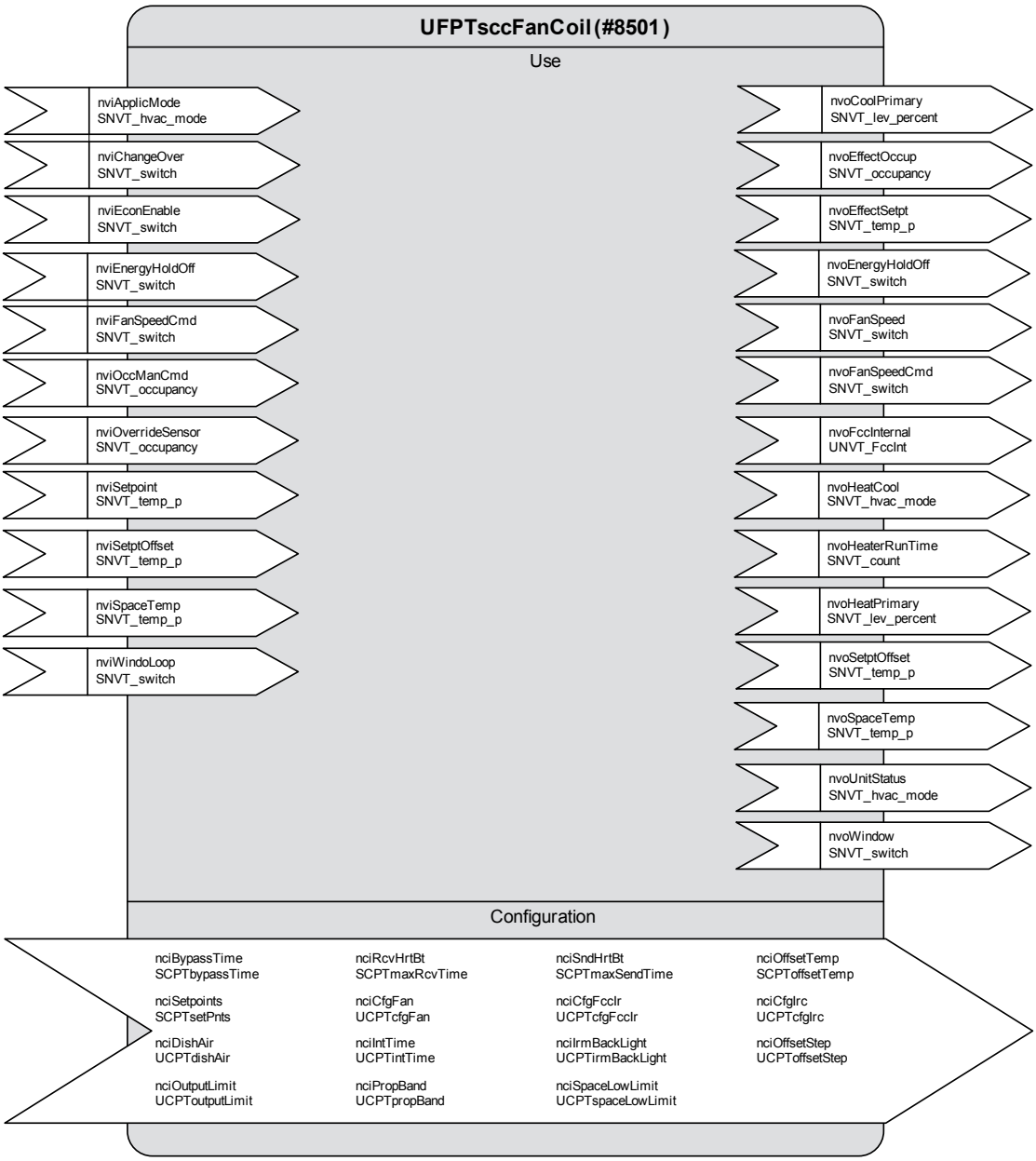
5.1 Node Object



Input variable	Type	Description
nviRequest	SNVT_obj_request	Node status request. Specific manufacturer process on following requests: **RQ_CLEAR_RESET:** Reset time counter for electrical battery

Output variable	Type	Description
nvoFileDirectory	SNVT_address	---
nvoStatus	SNVT_obj_status	Node status. nvoStatus is sent as answer to nviRequest and after reset

5.2 sccFanCoil



Variables marked with a "*" are stored in EEPROM. Its integrity is ensured for a maximum of 10 000 writing cycles.

Configuration variable	Type	Description
nciByPassTime*	SCPTbypassTime SNVT_time_min	Duration in minutes of the forcing restart of the occupancy mode. 0: no restart Units: minute Default: 60 Range: 0..250
nciRcvHrtBt*	SCPTmaxRcvTime SNVT_time_sec	Heartbeat period applied for the receiving of variables: nviApplicMode nviSpaceTemp (act on nvoSpaceTemp as same time) *Unit: sec Default: 0 Range: 0..6553*
nciSndHrtBt*	SCPTmaxSendTime SNVT_time_sec	Heartbeat period only applies to variables: nvoOccManCmd nvoHeatCool nvoPrimContact nvoAuxContact *Units: sec Default: 0 Range: 0..6553*
nciOffsetTemp*	SCPToffsetTemp SNVT_temp_p	Measurement offset of the probe connected to the controller (analogue probe or digital room operation unit) for the room temperature. *Units: °C Default: 0 Range: -10...10*
nciSetpoints*	SCPTsetPnts SNVT_temp_setpt	Value of a warm or a cold set point according to the occupancy modes. It can have the following values : **.occupied_cool (23)** **.standby_cool (25)** **.unoccupied_cool (28)** **.occupied_heat (21)** **.standby_heat (19)** **.unoccupied_heat (16)** *Unit: °C* *Default: {23,00 25,00 28,00 21,00 19,00 16,00}* *Range : 10..35*

5

Configuration variable	Type	Description
nciCfgFan*	UCPTcfgFan UNVT_cfg_fan { Unsigned short ... ... **mode** ... **cfg** ... **override** ... **level1** ... **level2** ... **level3** ... **mini** ... **manuf1** }	Configurations of fan forcing and 3 fan speed start thresholds **.mode (0** Not used **.cfg (0)** Not used **.override (0)** Not used **.level1 (5)**: Request on regulation to go to Speed 1 Unit: % Range: 0..100 **.level2 (33)**: Request on regulation to go to Speed 2 Unit: % Range: 0..100 **.level3(66)**: Request on regulation to go to Speed 3 Unit: % Range: 0..100 **.mini (0)**: Not used **.manuf1 (0)**: Not used *Default: {0 0 0 5 33 66 0 0}*

5

Configuration variable	Type	Description
nciCfgFcc	UCPTcfgFcclr UNVT_CfgFcclr { Unsigned short ... ... **fcctype** ... **ValveCycleDur** ... **ElecCycleDur** ... **FanOp** ... **roomModuleType** ... **SensorSelect** ... **TempDisplay** ... **FccAuxContact** ... **FanOffDelay** ... **manuf** Signed short ... ... **Window** }	**.fcctype (two pipes e heater)**: See detailed description chapter "4.2.1. Regulator configurat.". For the default value, the controller is configured in 2 pipes cold – 2 wires mode. **.ValveCycleDur (20)**: Duration of a valve control cycle, warm or coldt *Unit: sec* *Range : 20..250* **.ElecCycleDur (120):** Duration of the electric heater control cycle, in seconds. *Unit: sec* *Range : 100..250* **.FanOp (0):** Number of the associated **0:** no forcing **1:** speed 1 in dead zone if occupied or standby **2:** speed 1 in dead zone **3:** no ventilation in warm mode **4:** no ventilation in cold mode **5:** no ventilation, regardless of the regulation mode. **.roomModuleType (4):** Type of the room operation unit **0:** digital (on RJ9 input) **1:** analogue (on screw terminals) **.sensorSelect (1):** Selection of the temperature source. **1:** analogue probe (screw terminals) **2:** room device (RJ9 connector) **.TempDisplay (0):** Value of the set point shift **0:** set point shift 1: room operation unit temperature **2:** actual calculated set point (with blink display) **3:** actual calculated set point **4:** actual calculated central set point **.FccAuxContact (0):** Duration of the post- 0: not used 1: not used 2: Change over 3: Dew point 4: not used 5: temperature sensor (in **nvoAuxSensor**) **.FanOffDelay (180):** Duration of the post-ventilation. Unit: sec Range : 10..255

5

Configuration variable	Type	Description
		.Window (1): Polarity of the window contact -1: always closed 0: normally open 1: normally closed **.manuf (0):** Not used *Default:* *{TWO_PIPES_E_HEATER 20 120 0 4 1 0 0 180 1 0}*
nciCfgIrc*	UCPTcfgFcclr UNVT_CfgFcclr { Unsigned short ... ... **ValveType** ... **HeatValveTime** ... **CoolValveTime** ... **P1Cfg** ... **P2Cfg** ... **L1Cfg** ... **K1Cfg** ... **roomModuleCfg** }	**.ValveType (0):** Valve type used on the PCD7.L611. **0:** PWM valve **1:** 3 points valve **2:** 0-10V valve **.HeatValveTime (0):** Valve cycle time for a 3 points valve. The default value 0 is processed as 30s. *Unit: sec* *Range : 10..255* **.CoolValveTime (0):** Not used **.P1Cfg (0):** Not used **.P2Cfg (0):** Not used **.L1Cfg (180):** Configuration of the input R. **0:** output for analogue room operation device (occupancy state) **1:** presence contact simulation, closed = presence (update the nvoPresence) **.K1Cfg (0):** Not used **.roomModuleCfg (255):** Not used *Default: { 0 0 0 0 0 0 0 255}*

5

Configuration variable	Type	Description
nciDischAir*	UCPTdischAir { Unsigned short **Type** SNVT_temp_p **Propband** SNVT_temp_p **Low** SNVT_temp_p **High** }	Configuration of the blow temperature limitation mode. Enabled only when auxiliary contact = probe (nciCfgFcc.FccAuxContact = 5). **.Type (0)** **0:** disabled **1:** low limit **2:** high limit **3:** low and high limit **.Propband (5):** Proportional band used. Unit: °C **.Low (12):** Value of the low limit. Unit: °C Range : 0..90 **.High (45):** Value of the high limit. Unit: °C Range : 0..90 *Default: {0 5,00 12,00 45,00 0}*
nciIntTime*	UCPTintTime SNVT_time_sec	Value of the integral time. Values under 20s are interpreted as 0 and disabled the integral. *Unit: sec Default: 600 Range: 60..6553*
ncOADamper*	UCPTirmBackLight SNVT_time_sec	Not used
nciOffsetStep*	UCPToffsetStep SNVT_temp_p	Value of the set point shift step. *Unit: hundredth of °C Default: 50 Range: 0..255*
nciOutputLimit*	UCPToutputLimit { SNVT_lev_percent ... ... **MinHeat** ... **MinCool** ... **MaxHeat** ... **MaxCool** }	Not used
nciPropBand*	UCPTpropBand SNVT_temp_p	Value of the proportional band used by the control loop. *Unit: °C Default : 5 Range: 2..20*
nciSpaceLowLim*	UCPTspaceLowLimit SNVT_temp_p	*Value of the no freeze temperature.* *Unit: °C Default: 8 Range: 0..20*

5

<table>
<tr><th>Configuration variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nviApplicMode</td><td>SNVT_hvac_mode</td><td>Operating mode of the controller.

-1, HVAC_NUL: not take in consideration.
0, HVAC_AUTO: the operating mode is determined by the controller.
1, HVAC_HEAT: warm mode forcing.
3, HVAC_COOL: cold mode forcing.
6, HVAC_OFF: controller stop, no freeze mode.
7, HVAC_TEST: test mode.
8, HVAC_EMERG_HEAT: warm emergency.

Default: HVAC_AUTO</td></tr>
<tr><td>nviChgOver</td><td>SNVT_switch</td><td>Change over mode command.

<table><tr><th>State</th><th>Value</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>Warm</td></tr><tr><td>1</td><td>100</td><td>Cold</td></tr></table>
Default: {0,0 0}

Remark: this variable is stored in EEPROM. So the number of write cycles is limited.</td></tr>
<tr><td>nviEconEnable</td><td>SNVT_switch</td><td>Energy saving management.

<table><tr><th>State</th><th>Value</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>Normal operation</td></tr><tr><td>1</td><td>0 – 100%</td><td>Percentage of electric heating limited to Value %</td></tr></table>
Default: {0,0 0}</td></tr>
<tr><td>nviEnergyHoldOff</td><td>SNVT_switch</td><td>Energy saving command. This command can be used with the window contact information.

<table><tr><th>State</th><th>Value</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>Normal operation</td></tr><tr><td>1</td><td>100</td><td>Stop controller</td></tr></table>
Default: {0,0 0}</td></tr>
</table>

<table>
<tr><th>Configuration variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nviFanSpeedCmd</td><td>SNVT_switch</td><td>Fan speed command.
5 states exist: stop, speed 1, speed 2, speed 3, AUTO. In the AUTO mode, the control loop determines the speed among the 4 other states.

<table>
<tr><th>State</th><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>NA</td><td>Stop</td></tr>
<tr><td>1</td><td>0</td><td>Stop</td></tr>
<tr><td>1</td><td>33</td><td>Speed 1</td></tr>
<tr><td>1</td><td>66</td><td>Speed 2</td></tr>
<tr><td>1</td><td>100</td><td>Speed 3</td></tr>
<tr><td>0xFF</td><td>NA</td><td>AUTO</td></tr>
</table>
The fan speed value is expressed in % of the maximum speed.

Default: {0,0 -1} : AUTO</td></tr>
<tr><td>nviOccManCmd</td><td>SNVT_occupancy</td><td>Occupancy mode of the controller. A modification of this value cancels the forcing value.
The value OC_NUL is processed as OC_OCCUPIED.

Default: OC_NUL
Range: OC_OCCUPIED, OC_UNOCCUPIED, OC_NUL, OC_STANDBY</td></tr>
<tr><td>nviOccSensor</td><td>SNVT_occupancy</td><td>Occupancy forcing command, from a room operation device or another control device (refer to nviOccManCmd also).

Default: OC_NUL
Range : OC_OCCUPIED, OC_UNOCCUPIED, OC_NUL</td></tr>
<tr><td>nviSetpoint</td><td>SNVT_temp_p</td><td>Sets the central set point (middle of dead zone). The regulator updates the heat and cool set points values.

Unit: °C Default: 327.67 Range : 5..40</td></tr>
<tr><td>nviSetptOffset</td><td>SNVT_temp_p</td><td>Value of the temperature offset for the temperature set point. This offset is taken into account only if the occupancy mode is set to occupied or standby.
The value 327.67 (0x7FFF) is not valid and is processed as 0.

Unit: °C Default: 0 Range: -10..10</td></tr>
<tr><td>nviSpaceTemp</td><td>SNVT_temp_p</td><td>Value of the measured room temperature used by the control loop, in °C. It is used if a binding on this variable exists.
The value 327.67 (0x7FFF) is interpreted as invalid value and is not processed.

Unit: °C Default: 327.67 Range: -10..50</td></tr>
</table>

Configuration variable	Type	Description
nviWindowLoop	SNVT_switch	Window contact information for looping when several controllers are present in the same room (refers to master / slave operation). State / Value / Description: 0 / 0 / Normal operation 1 / 100 / Stop controller *Default: {0,0 -1}*

5

Output variable	Type	Description
nvoCoolPrimary	SNVT_lev_percent	Cold valve opening value. *Unit: % Default: 0 Range: 0..100*
nvoEffectOccup	SNVT_occupancy	Actual occupancy state of the controller. Calculated from nviOccManCmd, nviOccSensor Value / Description: 0 / OC_OCCUPIED 1 / OC_UNOCCUPIED 3 / 0C_STANDBY *Default: {0,0 -1}*
nvoEffectSetpt	SNVT_temp_p	Value of the actual calculated temperature set point. *Unit: °C Default: 21°C*
nvoEnergyHoldOff	SNVT_switch	Energy saving command. This command can be used with the window contact information. State / Value / Description: 0 / 0 / Normal operation 1 / 100 / Control loop disabled (but freeze protection remains active) *Default: {0,0 0}*

<table>
<tr><th>Output variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nvoFanSpeed</td><td>SNVT_switch</td><td>Actual fan speed value.

<table>
<tr><th>State</th><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Stop</td></tr>
<tr><td>1</td><td>33</td><td>Speed 1</td></tr>
<tr><td>1</td><td>66</td><td>Speed 2</td></tr>
<tr><td>1</td><td>100</td><td>Speed 3</td></tr>
</table>
Value of the fan speed in % of the maximum speed.

Default: {0,0 0}</td></tr>
<tr><td>nvoFanSpeed-Cmd</td><td>SNVT_switch</td><td>Fan speed command. See nviFanSpeedCmd.

Default: {0,0 -1}</td></tr>
<tr><td>nvoFccInternal</td><td>UNVT_FccInt
{
Unsigned short ..
.. atmel
.. carte
Signed long ..
.. field1
.. field2
.. field3
Unsigned short ..
.. field4
}</td><td>Not used</td></tr>
<tr><td>nvoHeatCool</td><td>SNVT_hvac_mode</td><td>Actual operating mode of the controller.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>HVAC_HEAT</td></tr>
<tr><td>3</td><td>HVAC_COOL</td></tr>
<tr><td>6</td><td>HVAC_OFF</td></tr>
<tr><td>7</td><td>HVAC_TEST</td></tr>
<tr><td>8</td><td>HVAC_EMERG_HEAT</td></tr>
</table>
Default: HVAC_OFF</td></tr>
<tr><td>nvoHeaterRunT-ime*</td><td>SNVT_count</td><td>Electric heater operation time.
It is reset by sending the variable
nviRequest.object_request = RQ_CLEAR_RESET.

Unit: hour Default: 0 Range: 0..65535</td></tr>
<tr><td>nvoHeatPrimary</td><td>SNVT_lev_percent</td><td>Cold valve opening value.

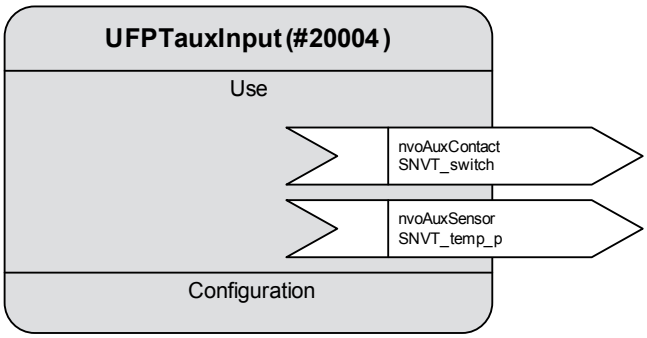
Unit: % Default: 0 Range: 0..100</td></tr>
<tr><td>nvoSetptOffset</td><td>SNVT_temp_p</td><td>Value of the temperature offset for the temperature set point. It is used for master / slave operation.

Unit: °C Default: 0 Range: -10..10</td></tr>
</table>

Output variable	Type	Description
nvoSpaceTemp	SNVT_temp_p	Value of the measured room temperature used by the control loop. *Unit: °C Default: 327.67 Range: -9.99°C..64.99°C*
nvoUnitStatus	SNVT_hvac_status	Controller status, comprising the following fields : **.mode (6):** the operating mode. See details in nvoHeatCool. **.heat_ouput_primary (0):** the warm valve operating value Unit: % Range: 0..100 **.heat_output_secondary (0):** the electric battery operating value Unit: % Range: 0..100 **.cool_output_primary (0):** the cold valve operating value Unit: % Range: 0..100 **.econ_output (0):** not used **.fan_output (0):** fan speed Unit: % Range: 0..100 **.in_alarm (0):** error (0: no error) *Default : {HVAC_OFF,0,0,0,0,0,0}*
nvoWindow	SNVT_switch	Window contact information used by the control loop. State / Value / Description 0 / 0 / Window closed 1 / 100 / Window open *Default : {0,0 -1}*

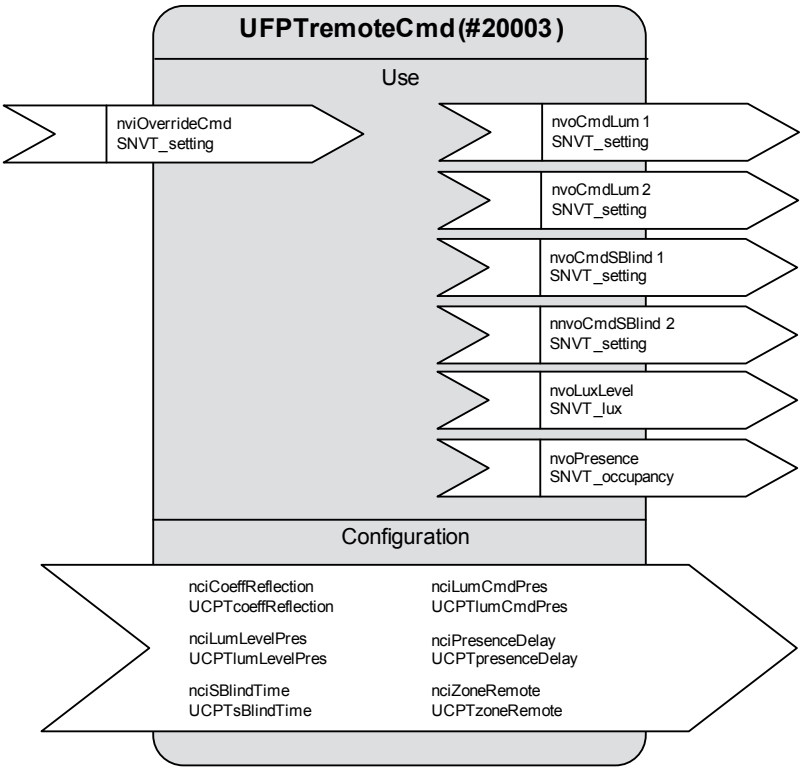
Variables marked with a "*" are stored in EEPROM. Its integrity is ensured for a maximum of 10 000 writing cycles.

5.3 AuxInput



Output variable	Type	Description
nvoAuxContact	SNVT_switch	State of the auxiliary contact, managed as normally closed (NC). State / Value / Description 0 / 0 / Closed contact 1 / 100 / Opened contact Valid when the contact is configured with a code different from 5.
nvoAuxSensor	SNVT_temp_p	*Blowing air temperature on auxiliary contact.* *Valid when the contact is configured with the code 5.*

5.4 Command



Variables marked with a "*" are stored in EEPROM. Its integrity is ensured for a maximum of 10 000 writing cycles.

Configuration variable	Type	Description
nciCoeffReflection*	UCPTcoeffReflection UNSIGNED_SHORT	Coefficient to calibrate the luminosity measurement, according to the room environment. $Actual\ luminosity = \frac{Measured\ luminosity\ x\ 100}{nciCoeffReflection}$ Units: % Default: 100 Range : 0..255

Configuration variable	Type	Description
nciLumCmd-Pres*	UCPTlumCmdPres { Unsigned short .. .. **lum1** .. **lum2** .. **Control** }	Parameters for the presence detection: **.lum1 (3):** effect of presence detection on light 1 **0:** no effect **1:** light is turned ON if presence **2:** light is turned OFF after timer has elapsed if no presence **.lum2 (3):** effect of presence detection on light 2 **0:** no effect **1:** light is turned ON if presence **2:** light is turned OFF after timer has elapsed if no presence **3:** light is turned ON and OFF (combination of 1 and 2) **.Control (0):** effect of presence detection on HVAC **0:** no effect **1:** occupancy forcing if presence (nviOccSensor is updated) *Default : {3 3 0}*
nciLumLevel-Pres*	UCPTlumLevelPres SNVT_lux	Luminosity level threshold on detection. *Unit: lux* *Default: 600* *Range: 0..65535*
nciPresenceDe-lay*	UCPTpresenceDelay SNVT_time_sec	Time during which the room is considered as occupied after a presence detection. After each detection, the timer is restarted. The value 0 is interpreted as 10 seconds. *Unit: sec* *Default: 600* *Range: 0..6553*
nciSBlindTime*	UCPTsBlindTime SNVT_time_sec	Timeout value for the UP or DOWN movement of the sunblind. *Unit: sec* *Default: 120* *Range: 1..6553*
nciZoneRemote*	UCPTzoneRemote SNVT_count	Number of the associated remote controller. **0:** the controller accepts the commands from any remote controller. **n (n≠0):** the controller accepts the commands from the remote controller with the number n only. *Unit: int* *Default: 0* *Range : 0..30*

<table>
<tr><th>Input variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nviOverrideCmd</td><td>SNVT_setting</td><td>Forcing of the light and sunblind commands.
The “rotation” field is never considered for forcing.

<table>
<tr><th>Function</th><th>Set-ting</th><th>Description</th></tr>
<tr><td>SET_UP</td><td>0</td><td>Forcing sun blinds to up*</td></tr>
<tr><td>SET_DOWN</td><td>0</td><td>Forcing sun blinds to down*</td></tr>
<tr><td>SET_ON</td><td>0</td><td>Forcing lights to ON</td></tr>
<tr><td>SET_OFF</td><td>0</td><td>Forcing lights to OFF</td></tr>
<tr><td>SET_NUL</td><td>0</td><td>Stop the last sun blinds forcing</td></tr>
</table>
*: Local command is disabled until the forcing is applied on the nviOverrideCmd.

Default: { SET_NUL 0,0 0,00}</td></tr>
</table>

<table>
<tr><th>Output variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nvoCmdLum1</td><td>SNVT_setting</td><td>Light 1 command, from room operation controller or forcing on the nviOverrideCmd.

<table>
<tr><th>Function</th><th>Value</th><th>Description</th></tr>
<tr><td>SET_ON</td><td>0</td><td>Forcing lights to ON</td></tr>
<tr><td>SET_OFF</td><td>0</td><td>Forcing lights to OFF</td></tr>
<tr><td>SET_STATE</td><td>X</td><td>Forcing lights level to X%</td></tr>
<tr><td>SET_STOP</td><td>0</td><td>Stop the last action
(for a dimming lamp)</td></tr>
</table>
Default: { SET_OFF 0,0 0,00}</td></tr>
<tr><td>nvoCmdLum2</td><td>SNVT_setting</td><td>Light 2 command, from room operation controller or forcing on the nviOverrideCmd.

<table>
<tr><th>Function</th><th>Value</th><th>Description</th></tr>
<tr><td>SET_ON</td><td>0</td><td>Forcing lights to ON</td></tr>
<tr><td>SET_OFF</td><td>0</td><td>Forcing lights to OFF</td></tr>
<tr><td>SET_STATE</td><td>X</td><td>Forcing lights level to X%</td></tr>
<tr><td>SET_STOP</td><td>0</td><td>Stop the last action
(for a dimming lamp)</td></tr>
</table>
Default: { SET_OFF 0,0 0,00}</td></tr>
</table>

<table>
<tr><th>Output variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nvoCmdSBlind1</td><td>SNVT_setting</td><td>Sunblind 1 command, from room operation controller or forcing on the nviOverrideCmd

<table>
<tr><th>Function</th><th>Value</th><th>Description</th></tr>
<tr><td>SET_ON</td><td>0</td><td>Forcing sun blinds to up</td></tr>
<tr><td>SET_OFF</td><td>0</td><td>Forcing sun blinds to down</td></tr>
<tr><td>SET_STATE</td><td>X</td><td>Stop the last action</td></tr>
</table>

Default: { SET_OFF 0,0 0,00}</td></tr>
<tr><td>nvoCmdSBlind2</td><td>SNVT_setting</td><td>Sunblind 2 command, from room operation controller or forcing on the nviOverrideCmd

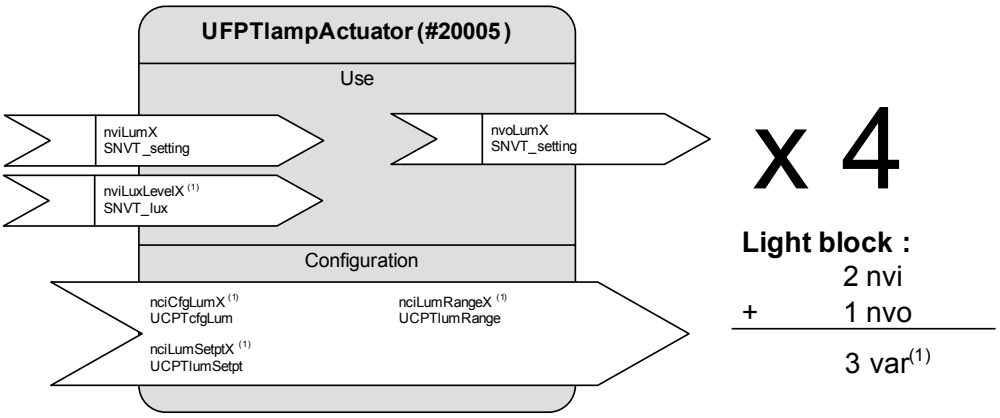
<table>
<tr><th>Function</th><th>Value</th><th>Description</th></tr>
<tr><td>SET_ON</td><td>0</td><td>Forcing sun blinds to up</td></tr>
<tr><td>SET_OFF</td><td>0</td><td>Forcing sun blinds to down</td></tr>
<tr><td>SET_STATE</td><td>X</td><td>Stop the last action</td></tr>
</table>

Default: { SET_OFF 0,0 0,00}</td></tr>
<tr><td>nvoLuxLevel</td><td>SNVT_lux</td><td>Luminosity level measured by the multi-sensor. This value is updated only if the new measured value differs by +/- 10 lux from the previous value.

Unit: lux Default: 0 Range: 0..1020</td></tr>
<tr><td>nvoPresence</td><td>SNVT_occupancy</td><td>State of the presence detector.

Default: OC_OCCUPIED
Range: OC_OCCUPIED, OC_UNOCCUPIED</td></tr>
</table>

5.5 LampX



(1) : These variables are available only on objects "lamp1" and "lamp2"

Variables marked with a "*" are stored in EEPROM. Its integrity is ensured for a maximum of 10 000 writing cycles.

Configuration variable	Type	Description
nciCfgLumX*	UCPTcfgLum UNSIGNED_SHORT	Definition of the lamp type for light group X **0:** ON/OFF lamp **1:** Dimming lamp *Default: 0*
nciLumRangeX*	UCPTlumRange { Unsigned short **start** Unsigned short **end** }	*Not used.*
nciLumSetptX*	UCPTlumSetpt SNVT_lux	*Not used.*

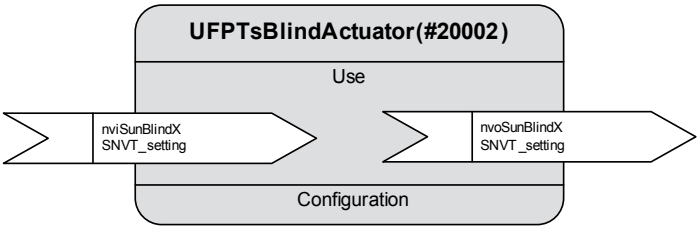
<table>
<tr><th>Input variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nviLumX</td><td>SNVT_setting</td><td>Lights group X command.
The “rotation” field is never considered.

<table>
<tr><th>Function</th><th>Set-ting</th><th>Description</th></tr>
<tr><td>SET_ON</td><td>0</td><td>Switched lights to ON
(or to 100% for dimming lamp)</td></tr>
<tr><td>SET_OFF</td><td>0</td><td>Switched lights to OFF
(or to 0% for dimming lamp)</td></tr>
<tr><td>SET_STATE</td><td>0</td><td>Switched lights in increase mode
(or turn off for a ON/OFF lamp)</td></tr>
<tr><td>SET_STATE</td><td>100</td><td>Switched lights in decrease mode
(or turn on for a ON/OFF lamp)</td></tr>
<tr><td>SET_STOP</td><td>0</td><td>Stop increase or decrease mode on dimming lights</td></tr>
</table>

Default: { SET_NUL 0,0 0,00}</td></tr>
<tr><td>nviLumLevelX</td><td>SNVT_lux</td><td>Not used.</td></tr>
</table>

Output variable	Type	Description
nvoLumX	SNVT_setting	State of lights group X, refer to the nviLumX. *Default: { SET_OFF 0,0 0,00}*

5.6 SBlindX



x 4

Light block :

1 nvi
\+ 1 nvo

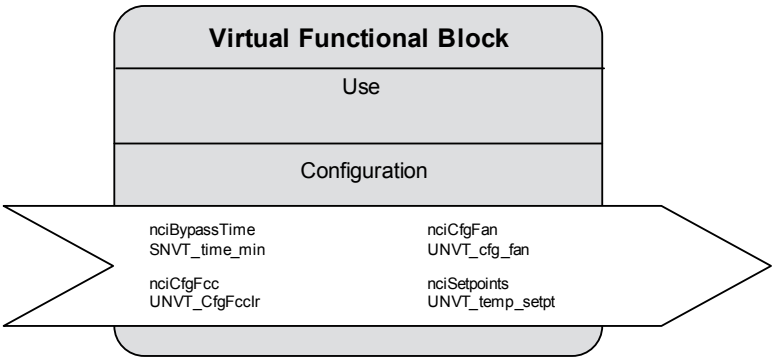
2 var

5

Input variable	Type	Description
nviSunBlindX	SNVT_setting	Lights group X command. The “setting” and the “rotation” field are never considered. Function / Description: SET_UP – Switched sun blinds in UP state SET_DOWN – Switched sun blinds in DOWN state SET_STOP – Stop up or down state on sun blinds *Default: { SET_NUL 0,0 0,00}*

Output variable	Type	Description
nvoSunBlindX	SNVT_setting	State of sun blinds group X, refer to the nviSunBlindX. *Default: { SET_OFF 0,0 0,00}*

5.7 Virtual Function Block

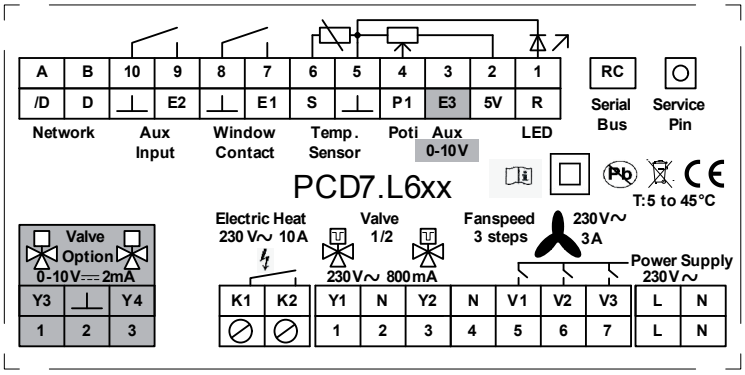


Variables marked with a "*" are stored in EEPROM. Its integrity is ensured for a maximum of 10 000 writing cycles.

Configuration variable	Type	Description
nciBypassTime*	SCPTbypassTime SNVT_time_min	Same as nciBypassTime in the sccFanCoil functional block but in configuration network variable version.
nciCfgFan*	UCPTcfgFan UNVT_cfg_fan	Same as nciCfgFan in the sccFanCoil functional block but in configuration network variable version.
nciCfgFcc*	UCPTcfgFcclr UNVT_CfgFcclr	Same as nciCfgFcc in the sccFanCoil functional block but in configuration network variable version.
nciSetpoints*	SCPTsetPnts SNVT_temp_setpt	Same as nciSetpoints in the sccFanCoil functional block but in configuration network variable version.

6 Technical data

	Terminal	Description
Power supply	L, N	230 VAC, 100 mA typical. No current to Triac outputs Y1/ Y2
Ouputs		
Fan	N, V1, V2, V3	230 VAC, 3A (AC3) max for direct control of a 3-step fan.
Valves	Y1, N, Y2	Triac outputs 230 VAC, 10...800 mA to control 2 valves with PWM signal or 3-point valve.
Valves	Y3, N, Y4	Constant voltage outputs 0…10 V, 2 mA max. to control 2 valves.
Electric heating	K1, K2	Floating relay contact 230 VAC, 2 kW max.
Inputs		
Window contact	E1, window contact	Digital input for floating contacts.
Additional input	E2, aux input	Additional digital input for floating contacts.
Temperature sensor	S, temp sensor	Input for a temperature sensor NTC 10 kΩ
Potentiometer	P1, poti	Input for a set point potentiometer, 10 kΩ linear
Voltage input	E3, aux 0...10V	Input for Fan speed forcing
Voltage output	5V	Voltage output 5 V to supply the potentiometer on terminal P1
Operating status	R, LED	Voltage output 5 V, 2 mA max. Comfort mode = HIGH (5 V), otherwise LOW (0 V)
Communication		
Communication	A, B	Connection for FTT-10 Lon Network
Serial bus	RC	Internal data bus for the extension modules and a digital room operation unit



A Annex

A.1 Icons

	In manuals, this symbol refers the reader to further information in this manual or other manuals or technical information documents. As a rule there is no direct link to such documents.
	This symbol warns the reader of the risk to components from electrostatic discharges caused by touch. **Recommendation:** Before coming into contact with electrical components, you should at least touch the Minus of the system (cabinet of PGU connector). It is better to use a grounding wrist strap with its cable permanently attached to the Minus of the system.
	This sign accompanies instructions that must always be followed.
	Explanations beside this sign are valid only for the Saia PCD® Classic series.
	Explanations beside this sign are valid only for the Saia PCD® xx7 series.

A

A.2 Order codes

	Type	Description	
Room controllers			
SBC Serial S-Net	PCD7.L600	230 VAC room controller with 2 Triac outputs, relay for electric heating and 3-step fan control	
	PCD7.L601	230 VAC room controller with 2 Triac outputs, 2 outputs 0…10 V, relays for electric heating and 3-step fan control	
	PCD7.L603	24 VAC room controller with 2 Triac outputs, 2 outputs 0…10 V, relays for electric heating with 3-step fan control (230 VAC)	
	PCD7.L604	230 VAC room controller with 2 Triac outputs, 2 outputs 0…10 V, incl. 24 VAC supply (7 W), relays for electric heating and 3-step fan control	
LonWorks®	PCD7.L610	230 VAC room controller with 2 Triac outputs, relay for electric heating and 3-step fan control	
	PCD7.L611	230 VAC room controller with 2 Triac outputs, 2 outputs 0…10 V, Relays for electric heating and 3-step fan control	
	PCD7.L614	230 VAC room controller with 2 Triac outputs, 2 outputs 0…10 V, incl. 24 VAC supply (7 W), relays for electric heating and 3-step fan control	
	PCD7.L615	Dual 230 VAC room controller for radiator/cooled ceiling combinations and VAV applications, 4 Triac outputs, 2 outputs 0…10 V, 2 relays for electric heating and independent interfaces for digital room control devices	
BacNet®	PCD7.L681	230 VAC room controller with 2 Triac outputs, 2 outputs 0…10 V, relays for electric heating and 3-step fan control	
Extension modules for light and shade			
	PCD7.L620	Extension module to control 2 light bars	
	PCD7.L621	Extension module to control 2 light bars and 1 blind motor	
	PCD7.L622	Extension module to control 3 blind motors	
	PCD7.L623	Extension module to control 2 blind motors 24 VAC with blade movement	
Room control units			
Analogue	PCD7.L630	Temperature sensor	
	PCD7.L631	Temperature sensor and set-point setting	
	PCD7.L632	Temperature sensor, set-point setting, presence sensor and LED	
Digital	PCD7.L640	Temperature sensor and set-point setting	
	PCD7.L641	Temperature sensor, set-point setting, presence sensor and LED	
	PCD7.L642	Temperature sensor, set-point setting, presence sensor, LED and fan control	
	PCD7.L643	Temperature sensor, function keys and LCD display for HeaVAC functions	
	PCD7.L644	Temperature sensor, function keys and LCD display for HeaVAC and light and shade functions	

A

	Type	Description
Remote control	PCD7.L660	IR remote control with LCD display, temperature sensor and wall mounting for fixed use
	PCD7.L661	IR receiver
	PCD7.L662	Wireless remote control with LCD display, temperature sensor and wall mounting for fixed use
	PCD7.L663	Wireless receiver
	PCD7.L664	Optional wall mounting for mobile use
	PCD7.L665	IR (infra-red) receiver with multi-sensor for temperature, presence and brightness for PCD7.L660
	PCD7.L666	IR and wireless receiver with multi-sensor for temperature, presence and brightness for PCD7.L660/L662

Expansion modules to connect third-party devices

	PCD7.L650	Expansion module to connect up to 8 external contacts for light&shade
	PCD7.L651	Wireless receiver to connect EnOcean room control devices

Accessories

	PCD7.L670	Connecting cable for room control units RJ9/RJ9, 10 m
	PCD7.L671	Connecting cable for room control units RJRJ11/cord, 10 m
	PCD7.L672	Connecting cable for room controller/extension modules RJ11/RJ9, 0.3 m
	PCD7.L673	Set of connecting cables for digital room control units, 3 × RJ9 and 1 × RJ11, length 11 m
	PCD7.L679	Manual control unit for room controller configuration

A

A.3 Contact

Saia-Burgess Controls AG
Bahnhofstrasse 18
3280 Murten
Switzerland

Phone ..... +41 26 672 72 72
Fax..... +41 26 672 74 99

Email support: ..... support@saia-pcd.com
Supportsite: ..... www.sbc-support.com
SBC site: ..... www.saia-pcd.com
International Represetatives &
SBC Sales Companies: ..... www.saia-pcd.com/contact

Postal address for returns from customers of the Swiss Sales office

Saia-Burgess Controls AG
Service Après-Vente
Bahnhofstrasse 18
3280 Murten
Switzerland

A