SAIA BURGESS CONTROLS

PCD7.L610 room controllers LON

Document 26-873 | Edition ENG04 | 2019-03-21

0

0 Contents

0

0.1 Document History

Date	Version	Changes	Remarks
2009-08-03	EN01		Adaptation of Comtec documentation
2010-05-11	EN01		Publication
2013-09-25	EN02		New logo and new company name
2015-12-04	ENG03	div.	Divers small changes
2019-03-21	ENG04	Chapt. 5.2	New definition parameter «ncInputCfg»

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.

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.

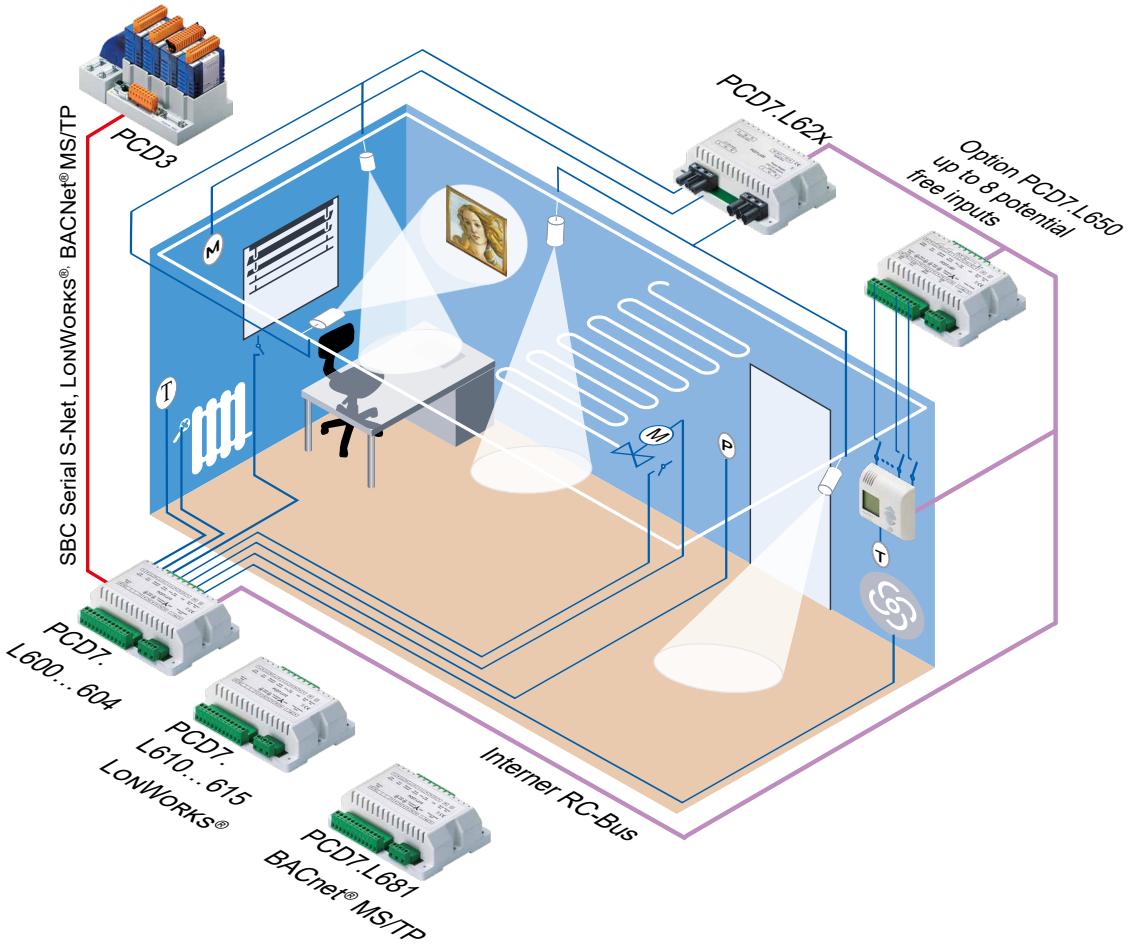
Manufacturer-independent room control units

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.

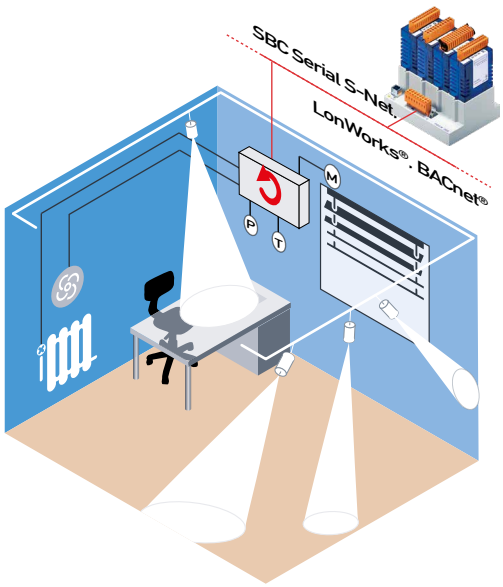
The default set-point setting of 21 °C can be modified by the set-point control (according to the device).

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.

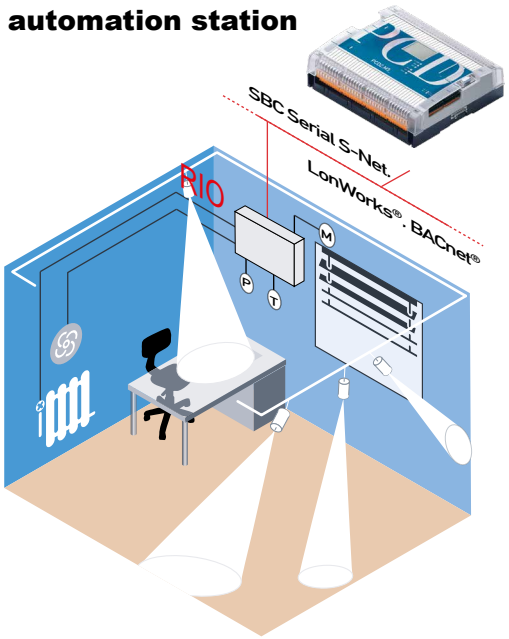
For parameterisation, there is a function object available in the library for every room controller type. In the case of open network connections, this is handled via network variables or network objects.



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 - 10 V	-	2x	2x with 24VAC supply	2x	2x
Fan 230 V	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	IP20			
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
Transmission rate	Saia PCD® 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 10 A
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

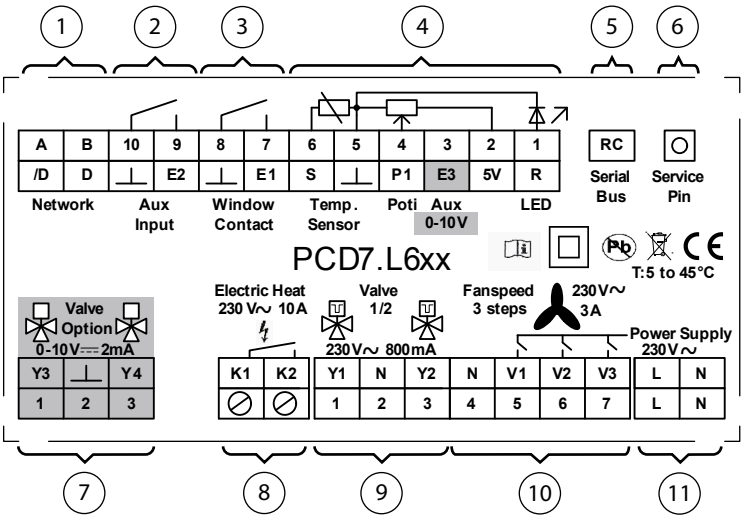
Program ID: 8F:FF:5B:55:01:04:04:60

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

Self documentation: PCD7L610 v100

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 - 5 V output - LED operation status output
5	serial bus (RJ9 connectors, either for room operation unit or extension devices)
6	push button (service pin)
7	on L610 not available
8	electric heater outputs 230 VAC / 10 A
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 Commissioning 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.

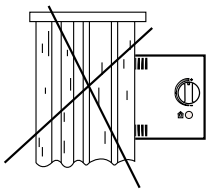
Please ensure that the limits specified in the technical data are not exceeded. Failure to do so may result in defects in the modules and the perpiherals connected to them. We can accept no responsibility for damage arising from improper deployment and use.

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.

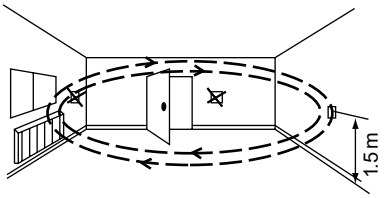
Please read this manual carefully before assembling and commissioning the modules. This manual contains instructions and warnings that must be observed to assure safe operation.

3.2 Assembly instructions

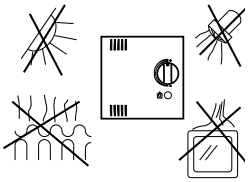
- The individual room controllers must only be installed and connected by an expert in accordance with the wiring diagram. Existing safety standards must also be observed.
- The individual room controller can only be used to regulate the temperature in dry, closed rooms. The maximum permissible relative 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.

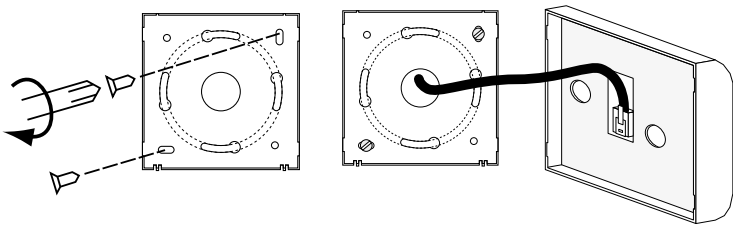


Do not install next to heat sources such as heaters, refrigerators, lamps etc.

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.

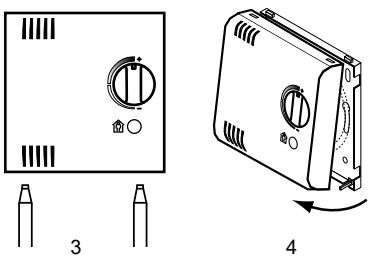
Mounting



1
Assembling the housing base

2
Wirin g to the device

Removal



3
Unlocking

4

4 Functionalities

4.1 Functional Block Overview

- 1× **Node object** (see Ch 5.1)

- 1× **sccFanCoil** object (see Ch 5.2)

- 1× **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

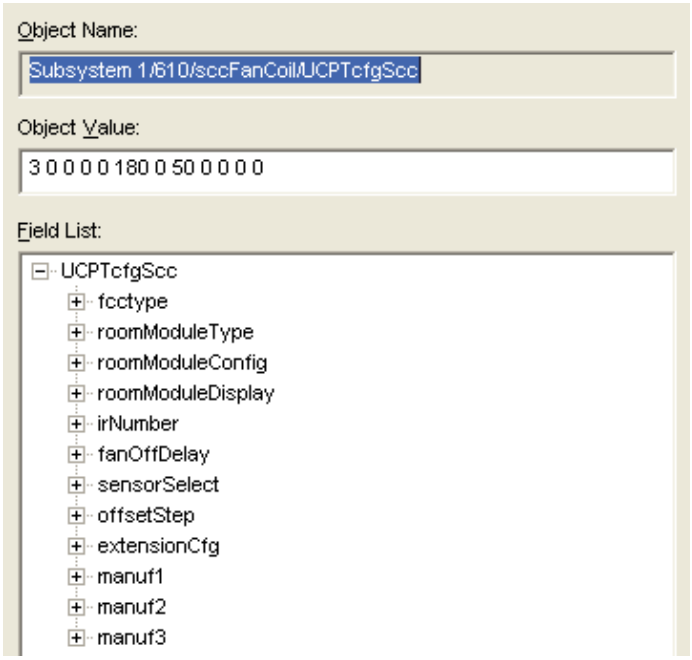
The PCD7.L610 can be used with a room local device to make available to the controller inputs needed for the regulation. The local device provides at the same time an interface for users to check and act on the process regulation (fitting of occupation, set point, ventilation…).

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?

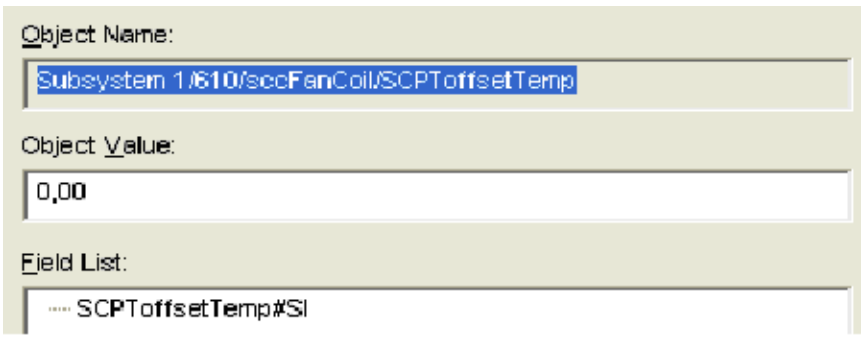
In the following description, only variables for room operation unit configuration are described.

nciCfgSrc	The main part of the configuration is done with this variable. It allows choosing the version of the device, analogue or digital, the room temperature origin and some other configuration especially for the offset.



.roomModuleType	To specify the technology of the room operation unit.	
	0	Digital, plugged on the serial bus (RJ9 connector)
	1	Analogue, plugged on analogue inputs S, P1, E3, 5 V and R. In this case, you need to configure also the sensor temperature origin to analogue sensor (see nciCfgSrc.sensorSelect). To check the technology used on the room operation unit, you can look to the serial bus connector. For a digital one it is equipped with a RJ9 connector, on the analogue version this one is a RJ11 connector.
.roomModuleConfig	Allow to not take in consideration value write in **nviOccManCmd** if the room operation unit is placed in unoccupied mode. This feature can be used to disable order from the BMS if users have placed the room in unoccupied mode (for time schedule for example).	
	0	Function disabled
	1	Function enabled.
.roomModuleDisplay	If the room operation unit is equipped with a LCD display, its displayed information can be changed with this variable.	
	0	Display of the fan speed.
	1	Display of the room operation unit temperature (updated each minute).
	2	Blink display of the actual calculated set point (with offset take in account).
	3	Display of the room temperature used for the regulation.
	4	Displays the actual calculated set point (with a considered shift).

nciCfgSrc	The main part of the configuration is done with this variable. It allows choosing the version of the device, analogue or digital, the room temperature origin and some other configuration especially for the offset.		
	.roomModuleDisplay	If the room operation unit is equipped with a LCD display,	
	.irNumber	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.
	.sensorSelect	To specify the origin of the room temperature used by the regulator. If the specified origin delivers an invalid temperature, the regulator tries to consider automatically a new one's on others sources (network or serial bus).	
		0	From the serial bus link.
		1	From an analogue input. This one can be used for the S input configured with the correct setting (see table 3) or with an analogue room operation unit.
	.offsetStep	To configure the value of one step offset. This value is in hundredth of °C and is bordered from 0 to 255 (0 °C to 2.5 °C).	

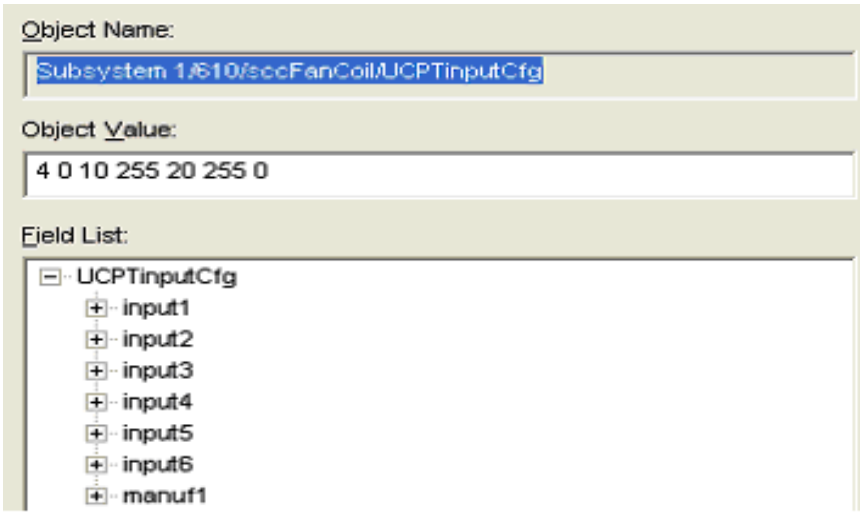
ncOffsetTemp (nciOffsetTemp)	Value of the offset applied by default on the temperature sensor selected with the nciCfgSrc.sensorSelect (analogue or digital sensor). This value is in °C and is bordered from -10 °C to 10 °C.
	

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	0…10 V	Internal code
Input 1	E1	**X**	**X**		Main contact, depending on the configuration
Input 2	E2	**X**			Auxiliary contact, depending on the configuration
Input 3	S		**X**		Sensor input for room temperature
Input 4	P1	**X**	**X**		Setpoint adjustment
Input 5	E3			**X**	0…10 V
Input 6	R	**X**			Led output of analogue room operation unit or digital input.

ncInputCfg	The functionality associated to each input can be configured with this configuration property.



.input1	Configuration of the input 1 (E1).
.input2	Configuration of the input 2 (E2).
.input3	Configuration of the input 3 (S).
.input4	Configuration of the input 4 (P1).
.input5	Configuration of the input 5 (E3).
.input6	Configuration of the input 6 (R).
.manuf1	Not used

Functions are described in next parts of this document (chapter 4.3. Functions).

Values for each of these parameters are described into the next table.

Function	ncInputCfg code	Updated variable	E2	E1	S	P1	E3	R
Not used	0xFF (255)							
Window	0	**nvoWindow**	X	X		X		X
Presence	1	**nvoPresence**	X	X		X		X
Dew point	2	**nvoDewSensor**	X	X		X		X
Change Over	3	**nvoChgOver**	X	X		X		X
Auxiliary contact (alarm status)	4	**nvoAlarm**	X					
Flow rate switch	5	**nvoFlowControl**	X	X		X		X
Auxiliary contact (information status)	6	**nvoAuxContact**	X					
Room temperature or return	10	**nvoSpaceTemp**			X			
Discharge air temperature	11	**nvoDischairTemp**	X		X			
Analog measurement 0-10 V	20	**nvoAnalogInput**					X	
Using with an analogue room device*								
(**nciCfgScc**.roomModuleType = 1)								
Set point shift		**nvoSetptOffset**				X		
Occupancy state output								X
Default value			4	0	10	0xFF	0xFF	0xFF

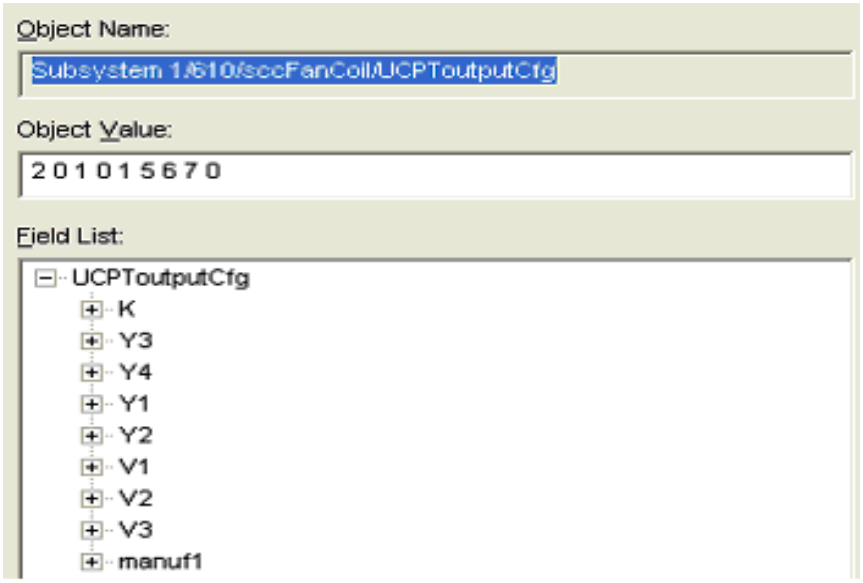
* Such as PCD7.L63x (At the moment when this document is written, references PCD7.L631 and PCD7.L632 are available).To use this kind of room operation unit, you need to configure the nciCfgScc.roomModuleType to 1.

4.2.3 Analogue Outputs

Like each input, each output can be configured to be associated to one function. To choose this function, you need to respect the type of the output describe in the next table.

Output	Pin	230 V	0-10 V	Switch	Internal description
K	K1-K2			**X**	Electric heater relay K
Y1	Y1	**X**			Triac Y1
Y2	Y2	**X**			Triac Y2
V1	V1	**X**			Fan speed V1
V2	V2	**X**			Fan speed V2
V3	V3	**X**			Fan speed V3

ncOutputCfg	The functionality associated to each output can be configured with this configuration property.



K	Configuration of the output K (cf K1-K2)
Y3	Not used
Y4	Not used
Y1	Configuration of the output Y1
Y2	Configuration of the output Y2
V1	Configuration of the output V1
V2	Configuration of the output V2
V3	Configuration of the output V3

Values for each of these parameters are described into the next table.

Function	ncOutputCfg code	K	Y3	Y4	Y1	Y2	V1	V2	V3
Free	0xFF (255)	X			X	X	X	X	X
Control valve 1 – Reg1 – PWM*	0 or 1				X	X			
Control valve 2 – Reg2 – PWM*	0 or 1				X	X			
Electric heater	2	X							
Control valve 1: Reg1 – 3 points**	3				X				
Control valve 2: Reg2 – 3 points**	4				X				
Fan V1	5						X	X	X
Fan V2	6						X	X	X
Fan V3	7						X	X	X
Default value		2	0xFF	0xFF	0	1	5	6	7

*: Y1 or Y2 outputs with codes 0 and 1 configure both in PWM output (according to reg1 or reg2), with Y1 ≠ 3 or 4.

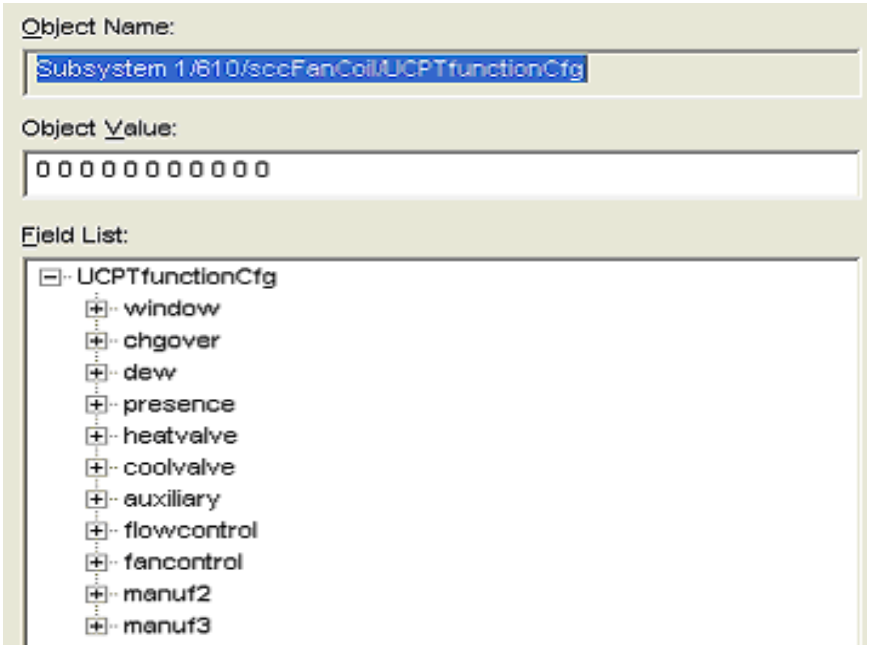
**: If Y1 output configure as 3 points output, the value for Y2 is not taken into account. This is because Y2 is compulsorily dedicated to a 3 points valve.

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.

To adapt the PCD7.L610 to all kind of installation, you can change the polarity of each input or output. These configurations are not directly applied to inputs or outputs, but to the function associated with it. This is done with the configuration property ncFunctionCfg.

ncFunctionCfg	Allow to configure the polarity of each function associate to an input or output.		
			
	.window	0: Normally open (NO) /	1: Normally closed (NC)
	.chgover	0: Open for warm /	1: Closed for warm
	.dew	0: Normally open (NO) /	1: Normally closed (NC)
	.presence	0: Open for occupied /	1: Closed for occupied
	.heatvalve	0: Normally closed (NC) /	1: Normally open (NO)
	.coolvalve	0: Normally closed (NC) /	1: Normally open (NO)
	.auxiliary	0: Normally open (NO) /	1: Normally closed (NC)
	.flowcontrol	0: Normally open (NO) /	1: Normally closed (NC)
	.fancontrol*	0: Direct control /	1: Inverse control
	.manuf2	Not used..manuf3 /	Not used

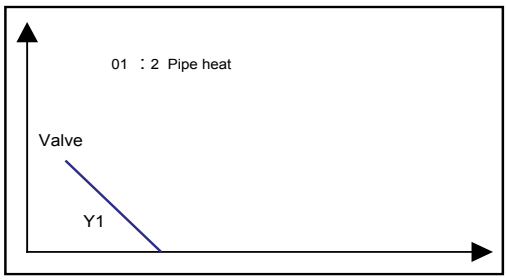
*: In direct control, the relay is activated to command the fan and deactivated to stop it. While for inverse control, the relay is not activated to command the fan and activated to stop it.

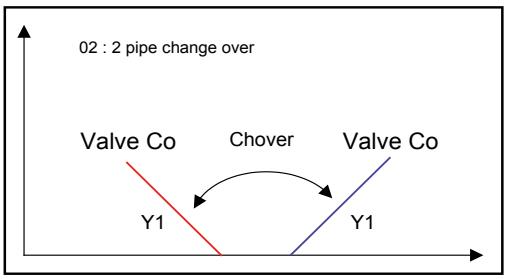
These parameters can only take values describe before.

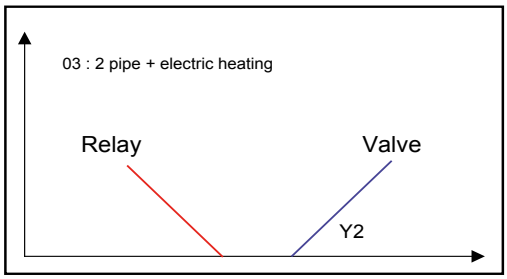
4.3 Application configuration

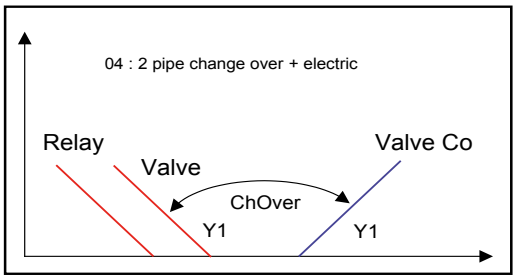
This chapter describe the configuration and the functioning of the HVAC regulator.

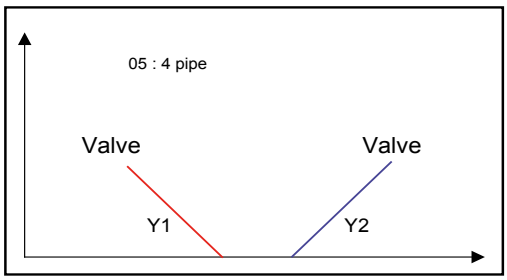
To be adaptable to much kind of installations, the application type needs to be setup in first. This is defined in the configuration variable **nciCfgSrc.type**. Supported application types are:

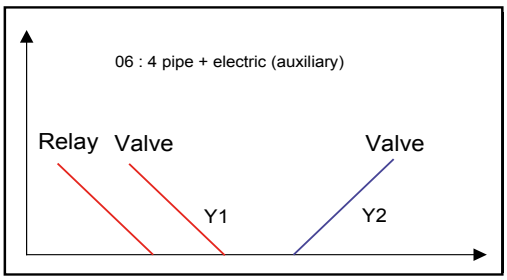


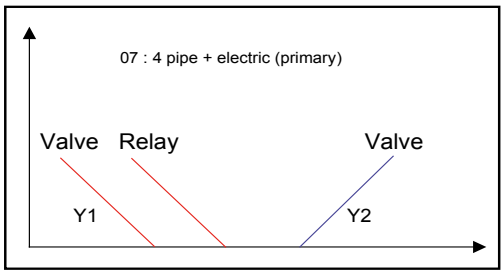


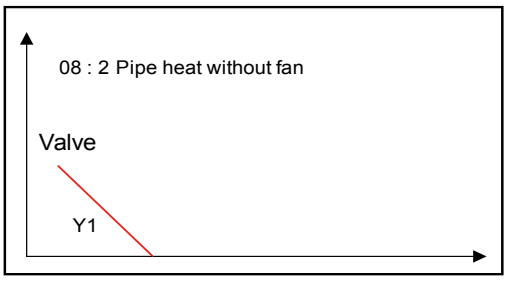


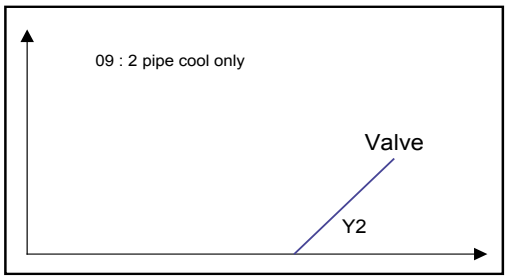


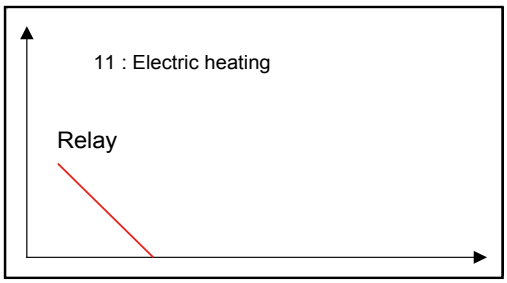












For change over state, valve is in cool mode when **nviChgOver**.state = 1 and in heat mode when **nviChgOver**.state = 0.

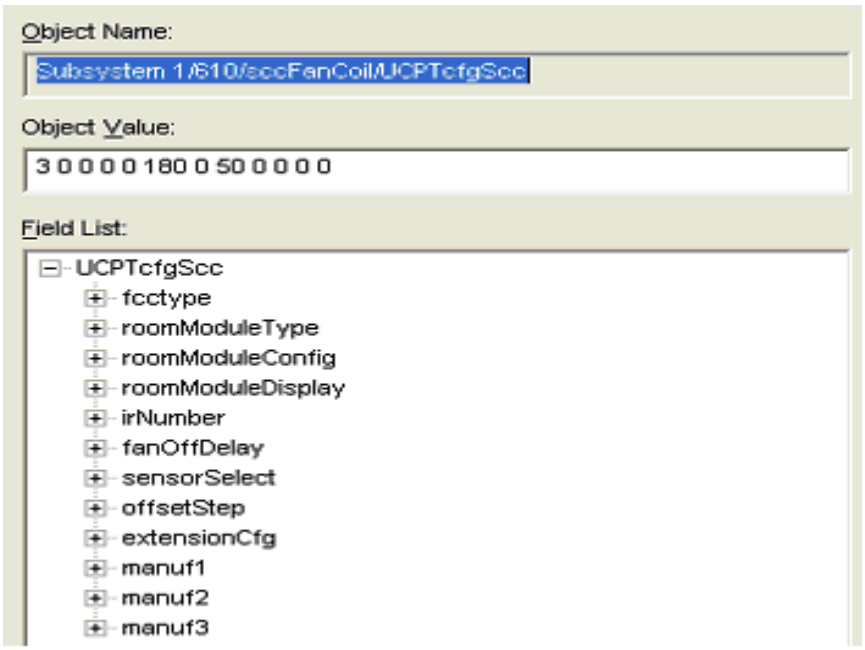
In next parts of this chapter, basic functions are described. These allow a quick setup of the controller by focusing only on them which are necessarily used for integration. For each functions, variable for configuration are described in first, followed by input and output variables for using this one. 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.

nciCfgSrc	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.

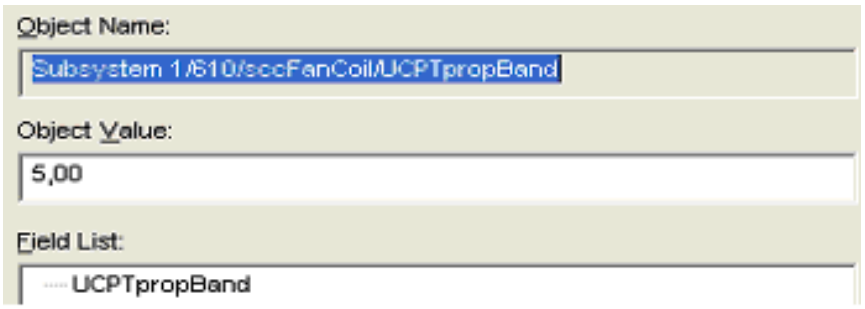


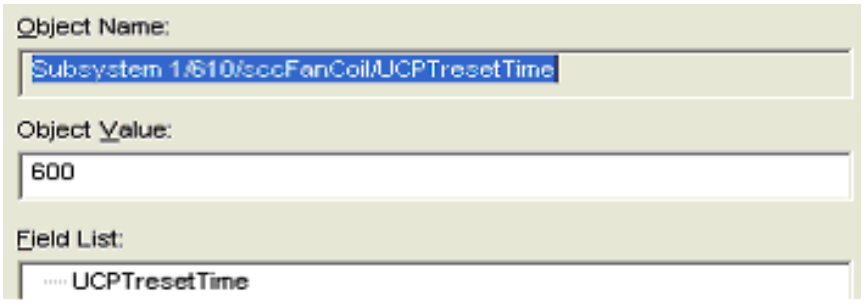
.fccType — To specify the installation type managed by the PCD7.L610.

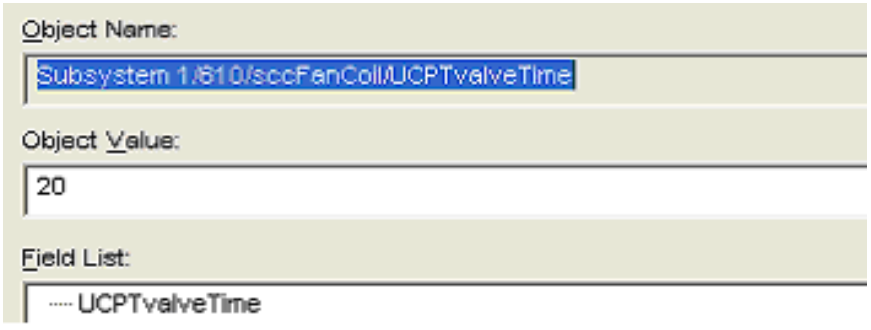
type	Description	C Over on Y1	Heat valve on Y1	Cool valve on Y2	Electric heater relay
01	2 pipe heat				
02	2 pipe change over				
03	2 pipe + electric heating				
04	2 pipe Change over + electric heating	primary			secondary
05	4 pipe				
06	4 pipe + electric heating (aux heat)		primary		secondary
07	4 pipe + electric heating (prim heat)		secondary		primary
08	2 pipe heat without fan				
09	2 pipe Cooling				
11	electric heating				

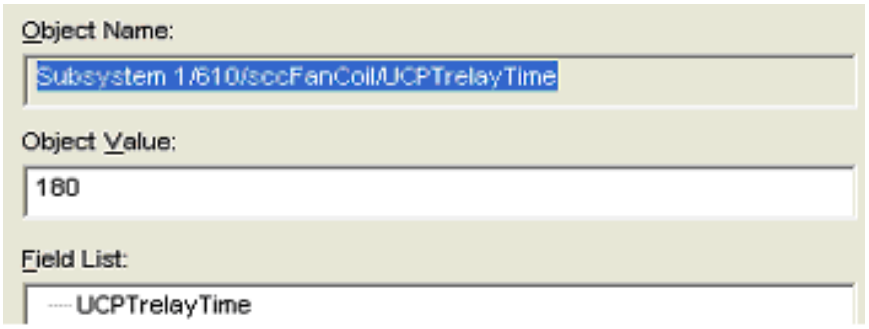
.fanOff Delay — Duration of the post ventilation function. It is used before to stop the fan, as well on a regulation order than on a user order. This value is in sec and is bordered from 10 s to 255s

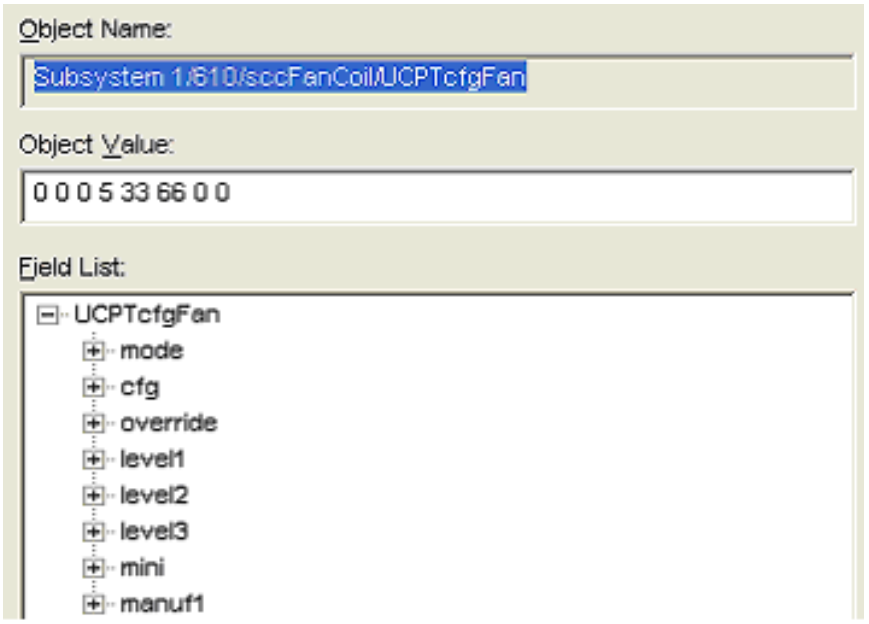
States for Reg1 and Reg2
can be viewed with **nvoOuputReg1** and **nvoOutputReg2**.

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.
	

ncResetTime	Value used in the PI regulator for the integral part. To disable the integral part, setup this parameter to 0s. This value is in sec and is bordered from 60s to 6553s.
	

ncValveTime	Time used as the valve cycle time. It is applied to valves configured in PWM or 3 points mode (See Chapter “4.1.3. Analogue outputs”).In the case of a 3 points valve, this time needs to match to the opening time of the valve. This value is in sec and is bordered from 20s to 250s.
	

ncRelayTime	Time used for the PWM cycle of the electric heater. This value is in sec and is bordered from 100s to 250s.
	

nciCfgFan	Configuration of the ventilation. For this part, we will focus only on the fan mode. Other parameters for advanced configuration will be described in chapter “4.3.2. Ventilation output control”.
	

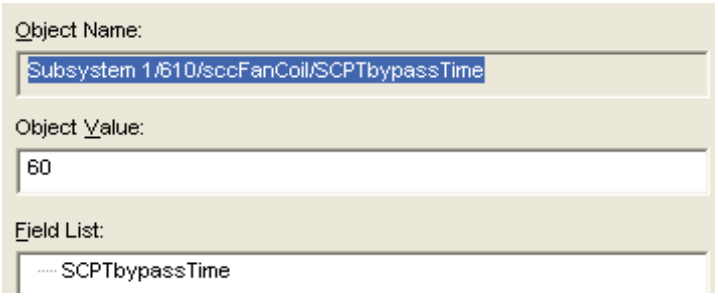
4.3.2 Occupancy mode management

The occupancy mode results from the synthesis of 3 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 **nviOverrideOcc** or with a room operation device. The forcing value is copied out to the **nvoOccManCmd**, and considered during the nciBypass-Time. After that, the command is reset to the **nviOccManCmd** value.
Presence signal	Presence detector can be plugged on the RJ9 link (with a PCD7.L665 for example) or on screw terminals (see chapter “4.1.2. Analogue inputs”). The state of the sensor is displayed by the **nvoPresence**.

The effective occupation state is given by the variable **nvoEffectOccup** after the computation of these 3 modes. Details about this computation are given by the next table:

Base mode	Forcing mode	Presence signal	Effective occupation
nviOccManCmd	**nviOverrideOcc** or local control device (**nvoOccManCmd**)	**nvoPresence**	**nvoEffectOccup**
OC_NUL	OC_NUL	OC_NUL	OC_OCCUPIED
OC_NUL	OC_NUL	OC_OCCUPIED	OC_OCCUPIED
OC_NUL	OC_NUL	OC_UNOCCUPIED	OC_UNOCCUPIED
OC_NUL	OC_OCCUPIED	No effect	OC_OCCUPIED
OC_NUL	OC_UNOCCUPIED	OC_OCCUPIED	OC_OCCUPIED
OC_NUL	OC_UNOCCUPIED	OC_UNOCCUPIED or OC_NUL	OC_UNOCCUPIED
OC_OCCUPIED	OC_NUL	No effect	OC_OCCUPIED
OC_OCCUPIED	OC_OCCUPIED	No effect	OC_OCCUPIED
OC_OCCUPIED	OC_UNOCCUPIED	No effect	OC_UNOCCUPIED
OC_UNOCCUPIED	OC_OCCUPIED	No effect	OC_OCCUPIED
OC_UNOCCUPIED	No effect	OC_ OCCUPIED	OC_OCCUPIED
OC_UNOCCUPIED	OC_UNOCCUPIED or OC_NUL	OC_UNOCCUPIED or OC_NUL	OC_UNOCCUPIED
OC_STANDBY	OC_OCCUPIED	No effect	OC_OCCUPIED
OC_STANDBY	No effect	OC_ OCCUPIED	OC_OCCUPIED
OC_STANDBY	OC_UNOCCUPIED or OC_NUL	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 0 min to 255min.
	

nviOccManCmd	The nviOccManCmd variable defines the operating mode sent by the BMS. Each time a new value of the **nviOccManCmd** variable is received: - **nviOverrideOcc** is cancelled (needs the value to be resent). - **nviOccManCmd** is copied on **nvoOccManCmd** (if OC_NUL, update with OC_OCCUPIED). - Forcing of the ventilation in Auto mode.

nviOverrideOcc	The **nviOverrideOcc** variable is used to force the occupation state by the network. This action can also be done with a room operation unit but the order is not written in this variable. In all cases, forcing value is copied into the **nvoOccManCmd**. In case option “Unoccupied disabled from room operation unit: nciCfgSrc. roomModuleConfig=1”, if user selects unoccupied mode from local device, the controller doesn’t consider information coming from the **nviOccManCmd** variable.

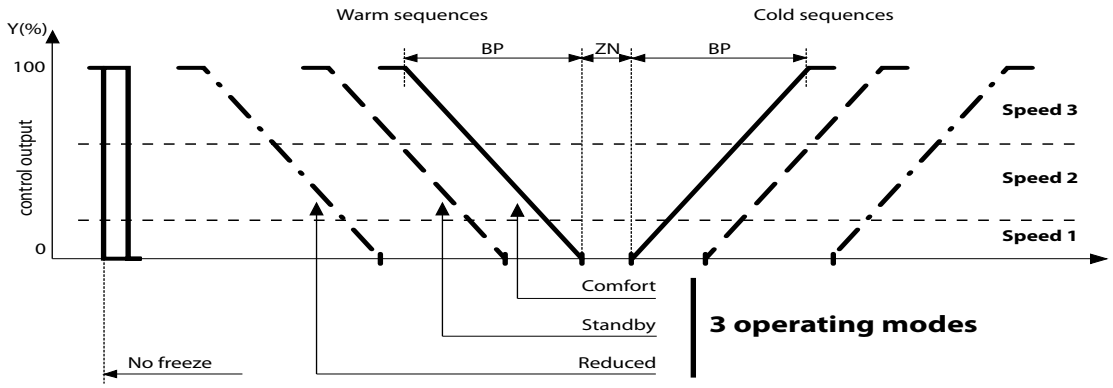
nvoEffectOccup	Effective occupancy state of the controller used for the regulation. At power-up **nvoEffectOccup** is set to OC_OCCUPIED, due to states of **nviOverrideOcc**, **nvoOccManCmd** and **nvoPresence**.

nvoOccManCmd	Occupation state from the BMS. This value is over written when the occupancy state is forced by a room operation unit or by the **nviOverrideOcc**.

nvoPresence	The detection sets **nvoPresence** to OC_OCCUPIED during 5 minutes. Then **nvoPresence** is reset to OC_UNOCCUPIED. At power-up **nvoPresence** is set to OC_NUL.

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 4 different cases for the computation of the set point. The three first are “Comfort”, “Standby” and “Reduced” mode and the last one is a specific application mode dedicated to the morning heating.

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

The last mode for morning heating is activated by forcing the application mode with the **nviApplicMode** set to the correct value, **nviApplicMode** = HVAC_MRNG_WRMUP.

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

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

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

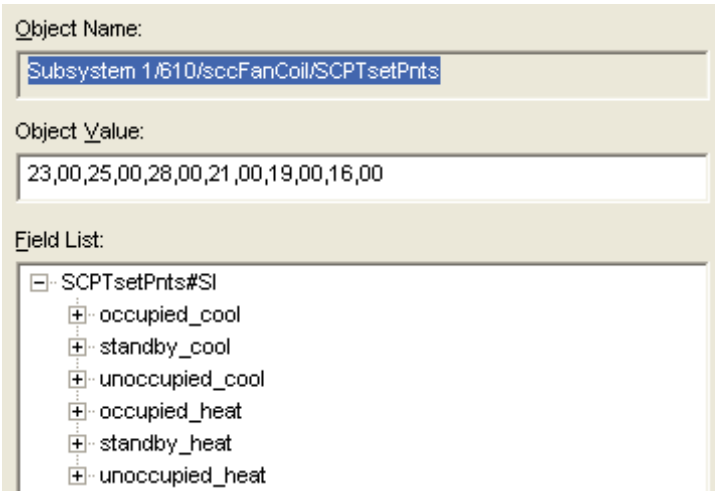
Unoccupied (nvoEffectOccup = OC_UNOCCUPIED) mode

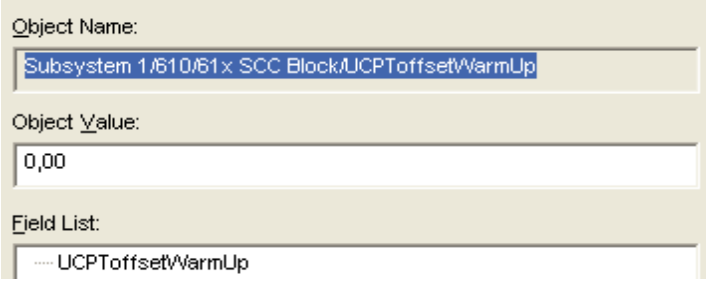
- Warm set point = **nciSetpoints**.unoccupied_heat
- Cold set point = **nciSetpoints**.unoccupied_cool

Specific morning heating application (nviApplicMode = HVAC_MRNG_WRMUP)

- Warm set point = **nciSetpoints**.occupied_heat + **nvoSetptOffset** + BMSOffset + ncOffsetWarmUp
- Cold set point: Not used, regulator in HVAC_MRNG_WRMUP supports only the HVAC_HEAT application mode.

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.
	

ncOffsetWarmUp	Offset value for the pre-warning mode HVAC_MRNG_WRMUP. This value is in °C and is bordered from -10 °C to 15 °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. 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 1Temperature sensor configured by default for the controller in the nciCfg-Src.SensorSelect (see chapter 4.1.1 Room operation unit).

2 f 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	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_MRNG_WRMUP (2):	over-warming mode.
	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
	HVAC_FAN_ONLY (9):	fan only mode
	All others: warm mode forcing.	

nviEnergyHoldOff	Used to enable or stop the control loop (refer to table 10).

nvoEnergyHoldOff	State of the control loop (refer chapter 4.4.4).

nvoHeatCool	State of the effective application mode of the controller.

nvoOutputReg1	State of output used as Reg1 (refer to table 6).

nvoOutputReg2	State of output used as Reg2 (refer to table 6).

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.

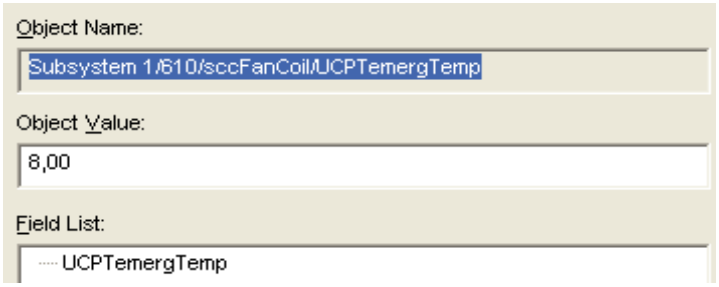
4.4.1 Frost-guard mode

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. When the regulation is in the dead zone, ventilation is stopped. If you want to force the speed fan to 1 in this zone used the **nciCfgFan.override**.

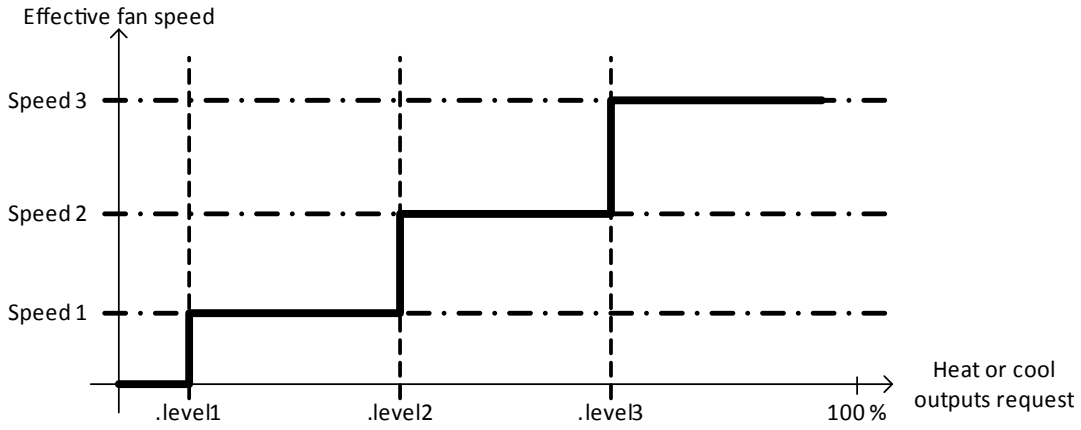
For forced mode, orders can be sent from the room operation unit or by the network, using the **nviFanSpeedCmd**. Orders sent can be viewed in the **nvoFanSpeedCmd** whereas the effective state of the fan is placed in the **nvoFanSpeed**.

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, 10 s.

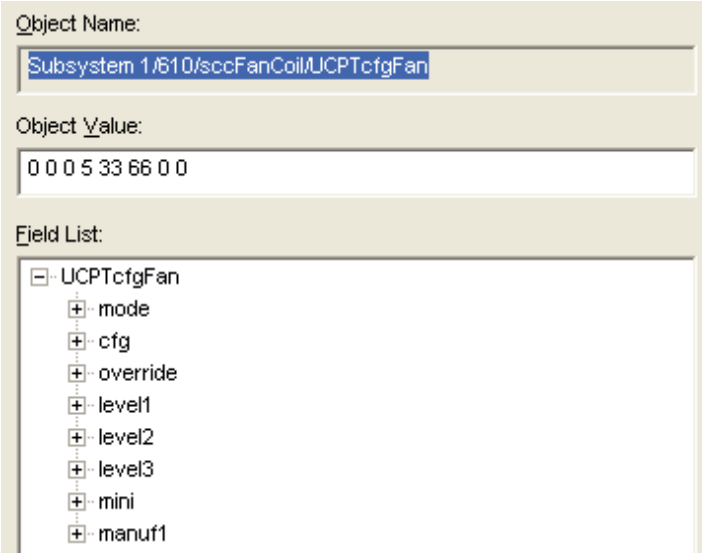
If the ventilation is forced to stop with the room operation unit, the regulation is stopped at the same time, unless it is in frost guard mode. The post ventilation time is still kept before to force the ventilation to stop.

It is possible to configure the controller for forcing speed 1 minimum even if the regulation is not in the dead zone by using the **nciCfgFan.override**. It is also possible to force ventilation to stop according to the application mode (heating and cooling) with the variable **nciCfgFan.cfg**.

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 type and its use.



.mode	Ventilator type managed by this room controller.	
	0	3 speed fan mode.
	1	Variable speed fan mode.
.cfg	Used to disable the ventilation according to the application mode.	
	0	Normal operation.
	1	Ventilation is always forced to stop..
	2	Ventilation is forced to stop in warm mode
	3	Ventilation is forced to stop in cold mode.
.override	Used to configure a minimal fan speed, depending on which override mode chosen in the next list.	
	0	No forcing.
	1	Mini V1 occupied and standby.
	2	Mini V1 occupied and standby unless stop is forced by local control device or by the **nviFanSpeedCmd**.
	3	Mini V1.
	4	Same as “2” but if ventilation is forced to stop and occupancy state is Unoccupied, ventilation is restarted in speed 1 for 5 minutes every 2 hours.
.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.	

nviFanSpeedCmd	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.

nvoFanSpeed	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.

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**.

nviChgOver	To forced the state of the change over.

nvoChgOver	Display the state of the change over. This variable is considered by the regulator to know in which case the Reg 1 can be used, for heating or for cooling.

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 **ncFunctionCfg**). In this case, the regulation is stopped (valve closed, fan and electric battery stopped) but frost guard mode is still active.

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.

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

- Contacts plugged on the E2 or E1 input configured as window contact (see chapter "4.1.2. Analogue inputs"). In this case, the state of the contact is displayed by the **nvoWindow**. The device is intended to manage only one input configured for window contact. To manage several windows and doors, the installer needs to plug its contacts in series or parallel (depending of the normal state of contacts) on the correct input.
- By the Lon network with the **nviLoopWind** 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.

When using a bidirectional room operation unit with a LCD display, an alarm is displayed on the screen.„The detection of the open window can be done by two ways:" auf Seite 4-25

If E1 and E2 inputs are configured as window contact, both can stop the regulation. But, if one of these two windows is closed again the regulation is restart even if the other window is still open.

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.

nviLoopWind 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 Auxiliary contact

The auxiliary contact, designated by the name E2, can be used like an alarm input or simply to make available the contact state on the network. This configuration is done by the ncInputCfg (see chapter “4.1.2. Analogue inputs”).

The contact state is displayed by the **nvoAuxContact**.

For the alarm contact mode, the state of the contact is still updated in the **nvoAux-Contact** and copied at the same time in the **nvoAlarm**. When the regulator switch in alarm state, the regulation is stopped (valve closed, fan and electric battery stopped) but speed fan forcing and frost guard mode are still active.

nvoAlarm	Alarm state of the regulator. This variable is used by the auxiliary contact in alarm contact mode and by the flow control function too.
nvoAuxContact	State of the auxiliary contact, according to its polarity configuration..

nvoAlarm is based on the SNVT_switch format which is composed of 2 fields, “state” and “value”. This variable uses the SNVT_switch in concordance with the next table. These values can’t be used for the **nvoAuxContact** due to the contact polarity. So **nvoAuxContact** is free of interpretation, regarding to its configuration in the installation.

State	Value	Description
0	0	Alarm Off, normal operation
1	100	Alarm On, control loop disabled

4.4.6 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.

Two possibilities are offered by the regulator to get the dew point information:

- By analogue contacts configured in dew point mode. In this case, the contact polarity can be adjusted with the ncFunctionCfg (see chapter 4.1.2. Analogue inputs").
- By the network with the **nviDewSensor**.

nviDewSensor	Dew point state given from the network, principally used in master/slave mode. Only the **nviDewSensor**.state is used and take in account only if **nvoHeatCool**=HVAC_COOL.

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

nviDewSensor and **nvoDewSensor** 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	Normal operation
1	100	Dew detection active

4.4.7 Flow control

To prevent damage on the fan coil, a flow controller needs to be used. If the fan stop due to a mechanical failure or if the filter is block, the controller can't knows it unless with a flow controller. In the case, the regulation needs to be stop before to destroy your device.

The flow controller state is displayed by the **nvoFlowControl**. If it is activated for more than 2 minutes, the PCD7.L610 switch in alarm mode (**nvoAlarm**.state = 1) and the regulation is stop.

The alarm can be acknowledged only by a reset of the device or with the **nviRequest** using the object_request set to RQ_CLEAR_ALARM.

nvoAlarm	Dew point state given from the network, principally used in master/slave mode. Only the **nviDewSensor**.state is used and take in account only if **nvoHeatCool**=HVAC_COOL.

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

nvoAlarm and **nvoFlowControl** 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	Normal operation
1	100	Update **nvoAlarm** after 2 minutes

4.4.8 Actions of contacts on the process control loop

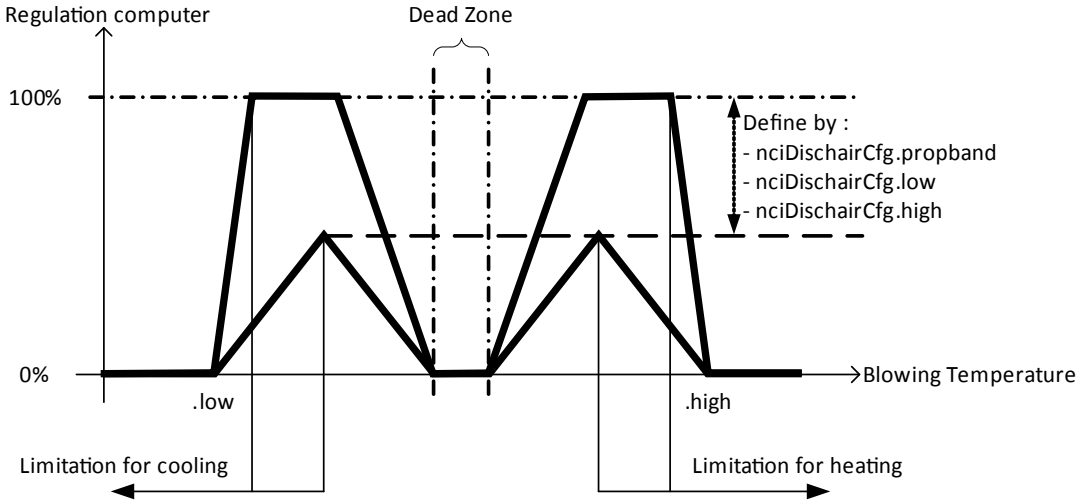
This table is a simply sum up of chapters “4.3.3. Change over” to “4.3.7. Flow control”.

Window	ncFunctionCfg.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	**nciCfgFccAuxContact**	**nvoAuxContact**	
contact “open”	3	{0 0}	Warm process control only – cold mode is stopped
contact “closed”	3	{1 100}	No effect
Change Over	ncFunctionCfg.chgover	**nvoChgOver**	
contact “open”	0	{0 0}	Warm mode
contact “open”	1	{1 100}	Cold mode
contact “closed”	0	{1 100}	Cold mode
contact “closed”	1	{0 0}	Warm mode
Auxiliary	ncFunctionCfg.auxiliary	**nvoAuxContact**	
contact “open”	0	{0 0}	No effect
contact “open”	1	{1 100}	No effect
contact “closed”	0	{1 100}	No effect
contact “closed”	1	{0 0}	No effect
FlowControl	ncFunctionCfg.flowcontrol	**nvoFlowCon-trol**	
contact “open”	0	{0 0}	No effect
contact “open”	1	{1 100}	Update nvoAlarm after a 2 minutes delay
contact “closed”	0	{1 100}	Update nvoAlarm after a 2 minutes delay
contact “closed”	1	{0 0}	No effect
Alarm	ncFunctionCfg. auxiliary	**nvoAlarm**	
contact “open”	0	{0 0}	No effect
contact “open”	1	{1 100}	Stop process control
contact “closed”	0	{1 100}	Stop process control
contact “closed”	1	{0 0}	No effect

The **nvoAlarm** variable is not maintained in case of update from auxiliary input configured in alarm mode (see chapter “4.3.5 Auxiliary contact”). But if **nvoAlarm** is activated by **nvoFlowControl**, the alarm is maintained and reinitialized by device reset or with **nviRequest** = 0,RQ_CLEAR_ALARM.

4.4.9 Blowing temperature limitation

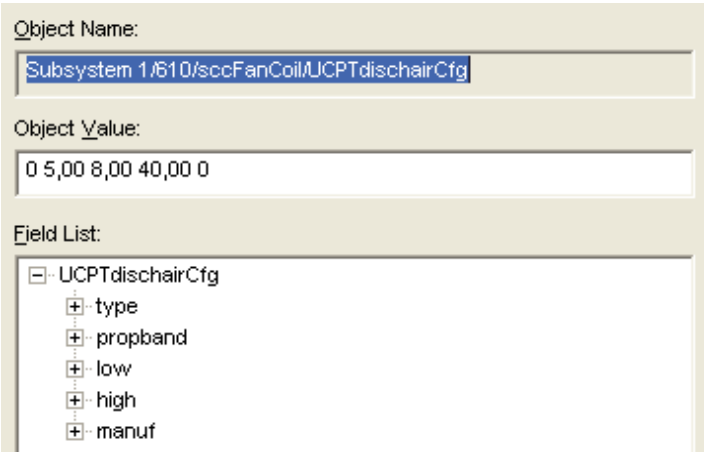
This function can be used to limit the temperature of the air blown by the device during the regulation. It permits to define two thresholds, one for warm air and one for the cold air. Each time the blowing temperature come up to these limits, valves or the electric heater are limited then stop when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

- For cooling limitation: To limit the cold air temperature, the low limit needs to be used (**nciDischairCfg.low**). The limitation will passed by 3 states during the decrease of the dish air temperature.
  - **nvoDischAirTemp > nciDischairCfg.low + nciDischairCfg.propband:**
    The regulation works normally, no limitation applied.
  - **nvoDischAirTemp < nciDischairCfg.low + nciDischairCfg.propband:**
    Limitation of the cold output proportionally to the difference with the low limit.
  - **nvoDischAirTemp < nciDischairCfg.low:**
    Cold output forced to 0 %
- For heating limitation: To limit the warm air temperature, the high limit needs to be used (**nciDischairCfg.high**).
  - **nvoDischAirTemp < nciDischairCfg.high - nciDischairCfg.propband:**
    The regulation works normally, no limitation applied.
  - **nvoDischAirTemp > nciDischairCfg.high - nciDischairCfg.propband:**
    Limitation of the warm output proportionally to the difference with the high limit
  - **nvoDischAirTemp > nciDischairCfg.high:**
    Warm output forced to 0 %

nciDischairCfg	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 90 °C.	
.high	Value of the high limit. This value is in °C and is bordered from 0 °C to 90 °C.	
.manuf1	Not used.	

ncOffsetDA	Offset use for the blowing air temperature sensor. This value is in °C and is bordered from -10 °C to 10 °C.

nvoDischAirTemp	Temperature measure by the discharge air temperature sensor. This value is in °C and is bordered from 0 °C to 90 °C.

4.4.10 Management of the electric heater

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. On the other hand, if the stop of the fan is caused by the forcing from the **nciCfgFan.cfg**, the electric battery is still used by the regulation.

The operation time for the electric heater is displayed by the **nvoElecCount**. This value is stored in the EEPROM memory of the device every 9 hours of functioning of the electric heater output. If a reset occurred, this value is reloaded from the EEPROM memory. To reset it, used the **nviRequest** with the value **nviRequest** = 0, RQ_OVERRIDE.

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

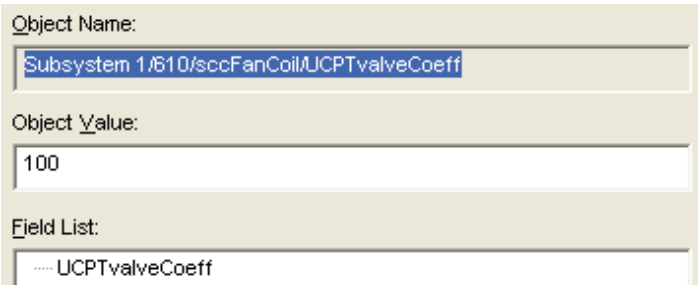
nvoAlarm and **nvoFlowControl** are based on the SNVT_switch format which is composed of 2 fields, “state” and “value”. These variables use the SNVT_switch in concordance with the next table.

4.4.11 Priority for action on valve output

In order to act with priority on valves as opposed to the ventilation, a multiplicative coefficient **ncValveCoeff** can be used. Updated values for outputs are limited to 100 %.

If you don’t want to use this action, simply keep the **ncValveCoeff** to its default value, 100 %.

This action can be done only on valve outputs, not on the electric heater.

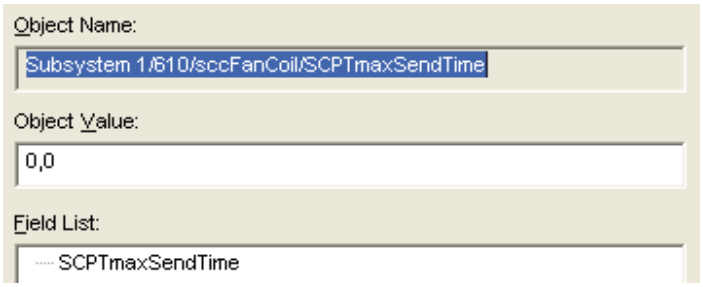
ncValveCoeff	Ratio to apply to heating and cooling outputs. This value is in percent and is bordered from 0 % to 250 %.
	

4.4.12 Forced variable propagation

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. This heartbeat is applied to:

- **nvoEffectOccup**
- **nvoHeatCool**
- **nvoWindow**
- **nvoAuxContact**

This function is mainly used in Master / Slave mode (see chapter "4.3.15 Master / Slave").

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

4.4.13 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.
- if **nviEconEnable.state** = 0xFF (Auto), load shedding if the temperature difference is lower than **nviEconEnable**.value (expressed in tenth of degree).

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 – %
0xFF	X	Electric heater stopped if temperature difference < 0,1.X°C	Tenth of °C
0xFF	0	Electric heater stopped if set point is reached	Tenth of °C

4.4.14 Direct control of outputs

It is possible to directly control the outputs Y1, Y2 and the contact of K1–K2, with the variables **nviOverY1**, **nviOverY2** and **nviOverRelay**, in 2 ways:

- Setting the code 0xFF into the configuration variable **nciOutputCfg** for the relevant output.
- Setting the variable **nviApplicMode** to HVAC_TEST, this disables the process control.

Direct control can't be used with PWM or 3 points output properties. They can only be forced to active or inactive, not to X%.

If an output is not used by the application type configuration (value of the nciCfgSrc. fccType), it isn't be forced in standard mode. Application mode needs to switch to HVAC_TEST or output needs to be configured as unused output too.

nviOverRelay	To force the state of electric heater contact.

nviOverY1	To force the state of the valve Y1

nviOverY2	To force the state of the valve Y2

nviOverY3	Not used.

nviOverY4	Not used.

nviOverRelay, **nviOverY1** and **nviOverY2** 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.

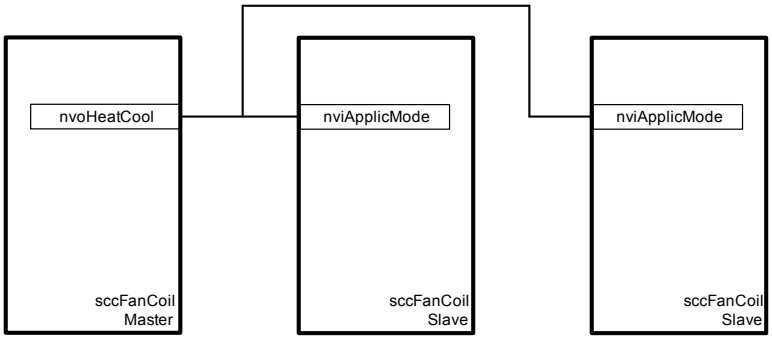
State	Value	Description
0	0	Output is disable
1	100	Output is enable

4.4.15 Master / Slave

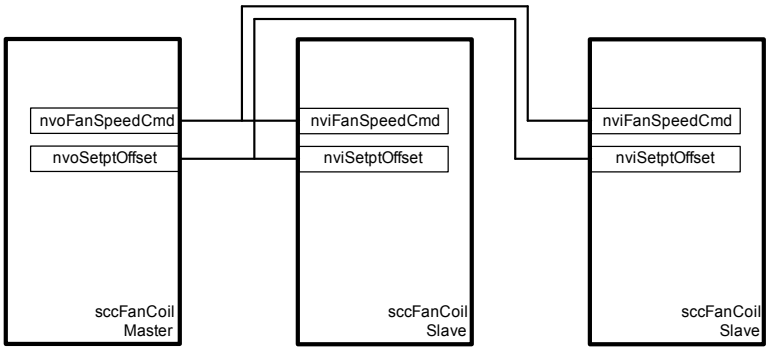
When several controllers are installed in the same room, it is necessary to have a consistency in the operation of these controllers. At this end, a controller will be defined as the « master » and this master will send at least the operating mode to the other controllers defined as the “slaves”:
nvoHeatCool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

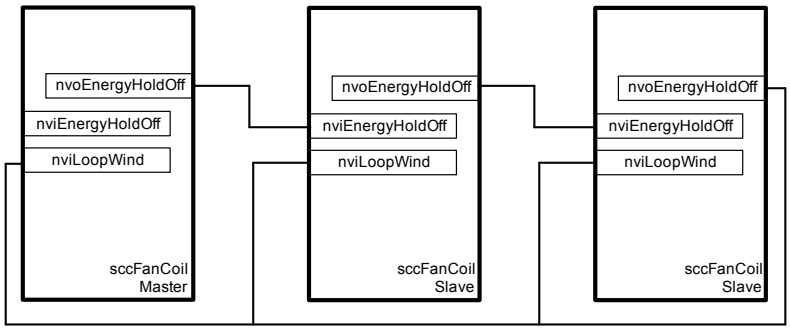
Application mode master/slave links:



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



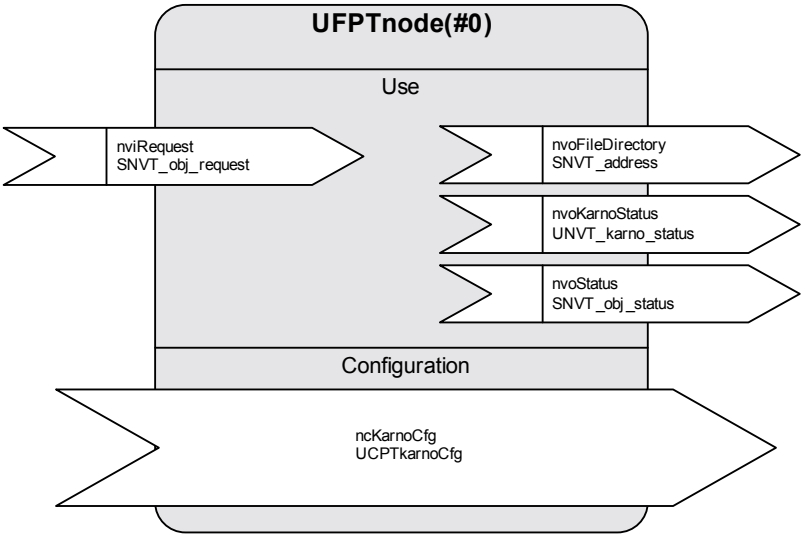
Window master/slave links:



5. Functional blocks and variables

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.

5.1. Node Object

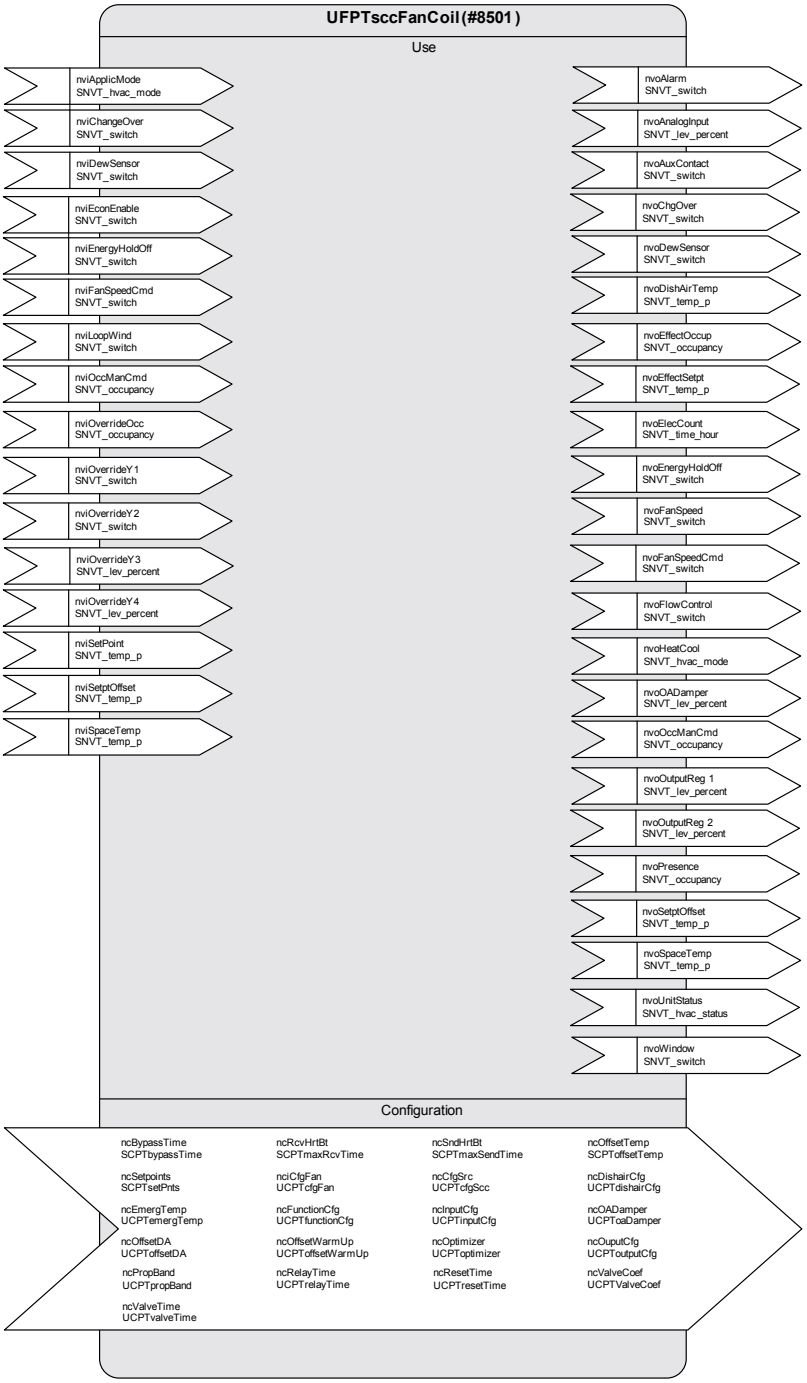


Configuration variable	Type	Description
ncKarnoCfg	UCPTkarnoCfg { Unsigned short ... ... **cfg** ... **number** ... **group** ... **maitre** ... **manuf0** }	Internal Configuration – not used and doesn't have to change. *Default : {2 0 0 0 0}*

Input variable	Type	Description
nviRequest	SNVT_obj_request	Node status request. Only Node (#0) requests are authorised with type RQ_NORMAL, RQ_UPDATE_STATUS, RQ_CLEAR_ALARM et RQ_REPORT_MASK. Specific manufacturer process on following requests: **RQ_PROGRAM:** Programming mode for Radio receiver on RJ9. **RQ_OVERRIDE:** Reset time counter for electrical battery.

Output variable	Type	Description
nvoFileDirectory	SNVT_address	
nvoKarnoStatus	UNVT_karno_status { Unsigned long ... ... **n_version** ... **a_minversion** ... **a_majversion** ... **manuf0** ... **manuf1** ... **manuf2** }	Manufacturer Variable Loaded NeuronChip application version. Minor Atmel version. Major Atmel version (Doesn't change, only for control). Not used. Not used. Not used.
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
ncByPassTime*	SCPTbypassTime SNVT_time_min	Duration in minutes of the forcing restart of the occupancy mode. 0: unlimited restart Units: minute Default: 60 Range: 0..250
nciRcvHrtBt*	SCPTmaxRcvTime SNVT_time_sec	Not used.
nciSndHrtBt*	SCPTmaxSendTime SNVT_time_sec	Heartbeat period only applies to variables: nvoOccManCmd nvoHeatCool nvoPrimContact nvoAuxContact Units: sec Default: 0 Range: 0..6553
ncOffsetTemp*	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
ncSetpoints*	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

Configuration variable	Type	Description
ncCfgFan*	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)** 0: 3-speed ventilation **.cfg (0)** **0**: normal **1**: no ventilation **2**: no ventilation in warm mode **3**: no ventilation in cold mode **.override (0)** **0**: no override **1**: if not unoccupied, fan speed 1 minimum **2**: if not unoccupied, fan speed 1 minimum, but stop is authorized **3**: fan speed 1 minimum, all modes **4**: same as 2, but systematic restart each 2 hours **.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
ncCfgSrc*	UCPTcfgScc UNVT_cfg_scc { Unsigned short ... ... **fcctype** ... **roomModuleType** ... **roomModuleConfig** ... **roomModule Display** ... **irNumber** ... **fanOffDelay** ... **sensorSelect** ... **offsetStep** ... **extensionCfg** ... **manuf1** ... **manuf2** ... **manuf3** }	**.fcctype (3)**: See detailed description chapter “4.2.1. Regulator configurat.”. For the default value, the controller is configured in 2 pipes cold – 2 wires mode. **.roomModuleType (0):** 0: digital room operation unit (on RJ9 input) 1: analogue room operation unit (on screw terminals) **.roomModuleConfig (0)**: Lock unoccupied mode from room operation unit 0: Function deactivated 1: Function enabled **.roomModuleDisplay (0):** Type of display of the room device. 0: fan speed 1: room operation unit temperature 2: actual calculated set point flashing 3: room temperature used for the regulation 4: indicator of actual calculated value (with a considered shift) **.irNumber (0):** 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 Range : 0..30 **.fanOffDelay (180):** Duration of the post-ventilation. Unit: sec Range : 10..255 **.sensorSelect (0):** Selection of the temperature source. 0: room operation unit (RJ9 connector) 1: analogue probe (screw terminals) **.offsetStep (50):** Value of the set point shift step. Unit: hundredth of °C Range: 0..255

5

Configuration variable	Type	Description
		.extensionCfg (0): Not used, reserved for further development **.manuf1 (0):** Not used **.manuf2 (0):** Not used **.manuf3 (0):** Not used *Default: {3 0 0 0 0 180 0 50 0 0 0 0}*
ncDischairCfg*	UCPTdischairCfg UNVT_dishair_cfg { Unsigned short **type** SNVT_temp_p **prop-band** SNVT_temp_p **low** SNVT_temp_p high Unsigned short **manuf** }	Configuration of the blow temperature limitation mode. **.type (0)** **0:** disabled **1:** low limit **2:** high limit **3:** low and high limit **.propband (5):** Proportional band used. Unit: °C **.low (8):** Value of the low limit. Unit: °C Range : 0..90 **.high (40):** Value of the high limit. Unit: °C Range : 0..90 **.manuf(0):** Not used *Default: {0 5,00 8,00 40,00 0}*
ncEmergTemp*	UCPTemergTemp SNVT_temp_p	Value of the no freeze temperature. Units: °C Default: 8 Range: 0 ... 20

5

Configuration variable	Type	Description
ncFunctionCfg*	UCPTfunctionCfg { Unsigned short ... ... **window** ... **chgover** ... **dew** ... **presence** ... **heatvalve** ... **coolvalve** ... **auxiliary** ... **flowcontrol** ... **fancontrol** ... **manuf2** ... **manuf3** }	Input / output polarity configuration **.window (0)** 0: normally open (NO) 1: normally closed (NC) **.chgover (0)** 0: open for warm 1: closed for warm **.dew (0)** 0: normally open (NO) 1: normally closed (NC) **.presence (0)** 0: open for occupied 1: closed for occupied **.heatvalve (0)** 0: normally closed (NC) 1: normally open (NO) **.coolvalve (0)** 0: normally closed (NC) 1: normally open (NO) **.auxiliary (0)** 0: normally open (NO) 1: normally close (NC) **.flowcontrol (0)** 0: normally open (NO) 1: normally closed (NC) **.fancontrol (0)** 0: direct control 1: reverse control **.manuf2(0):** Not used **.manuf3(0):** Not used *Default: {0 0 0 0 0 0 0 0 0 0}*

Configuration variable	Type	Description
ncInputCfg*	UCPTinputCfg { Unsigned short ... ... **input1** ... **input2** ... **input3** ... **input4** ... **input5** ... **input6** ... **manuf1** }	Input function configuration, see table 2 in Chapter "4.1.1. Inputs". **.input1 (4):** Configuration of the input E1 **.input2 (0):** Configuration of the input E2 **.input3 (10):** Configuration of the input S **.input4 (255):** Configuration of the input P1 **.input5 (20):** Configuration of the input E3 **.input6 (255):** Configuration of the input R **.manuf1 (0):** Not used *Default: {4 0 10 255 20 255 0}*
ncOADamper*	UCPToaDamper { Unsigned short ... ... **type** ... **cfg** ... level1 ... **level2** ... **manuf1** }	Not implemented.
ncOffsetDA*	UCPToffsetDA SNVT_temp_p	Measurement offset of the probe connected to the controller for measuring the blow temperature. Units: °C Default: 0 Range: –10…10
ncOffsetWarmUp*	UCPToffsetWarmUp SNVT_temp_p	Set point offset for the pre-warming mode HVAC_MRNG_WRMUP, enabled by nviApplicMode. Units: °C Default: 0 Range: –10…10
ncOptimizer*	UCPToptimizer { Unsigned short ... ... **mode** Unsigned long **timer** Unsigned short **heat-prop** Unsigned short **cool-prop** SNVT_temp_p **offset** Unsigned short **manuf1** Unsigned short **manuf2** }	Not used, reserved for further development

Configuration variable	Type	Description
ncOutputCfg*	UCPToutputCfg { Unsigned short ... ... **K** ... **Y3** ... **Y4** ... **Y1** ... **Y2** ... **V1** ... **V2** ... **V3** ... **manuf1** }	Output function configuration, see table 4 in capter “4.1.2. Outputs” **.K (2):** Configuration of outputs K1-K2. **.Y3 (0):** Not used **.Y4 (1):** Not used **.Y1 (0):** Configuration of the output Y1. **.Y2 (1):** Configuration of the output Y2. **.fan1 (5):** Configuration of the output V1. **.fan2 (6):** Configuration of the output V2. **.fan3 (7):** Configuration of the output V3. **.manuf1 (0):** Not used *Default: { 2 0 1 0 1 5 6 7 0}*
ncPropBand*	UCPTpropBand SNVT_temp_p	Value of the proportional band used by the control loop. Units: °C Default: 5 Range: 2 .. 20
ncRelayTime*	UCPTrelayTime Unsigned short	Value of the PWM cycle time of the K relay Units: sec Default: 100 Range: 100 .. 250
ncResetTime*	UCPTresetTime SNVT_time_sec	Value of the integral time. The value 0 disables the integral. Units: sec Default: 600 Range: 60 .. 6553
ncValveCoeff*	UCPTvalveCoeff Unsigned short	Coefficient to apply to the valve control output. Units: % Default: 100 Range: 0 .. 250
ncValveTime*	UCPTvalveTime Unsigned short	Value of the valve cycle time for PWM or 3 points valves Units: sec Default: 20 Range: 20 .. 250

<table>
<tr><th>Input 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.
2, HVAC_MRNG_WRMUP: over-warming mode.
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.
9, HVAC_FAN_ONLY: fan only mode
All others: warm mode forcing.

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

<table><tr><td>State</td><td>Value</td><td>Description</td></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>nviDewSensor</td><td>SNVT_switch</td><td>State of the dew point sensor. Only the nviDewSensor.state is used and take in account only if nvoHeatCool=HVAC_COOL.

<table><tr><td>State</td><td>Value</td><td>Description</td></tr><tr><td>0</td><td>0</td><td>Normal operation</td></tr><tr><td>1</td><td>100</td><td>Force cooling output to 0 %</td></tr><tr><td></td><td></td><td></td></tr></table>Default: {0,0 0}</td></tr>
<tr><td>nviEconEnable</td><td>SNVT_switch</td><td>Energy saving management.

<table><tr><td>State</td><td>Value</td><td>Description</td></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><tr><td>0xFF</td><td>0 – 255 °C</td><td>Electric heater stopped if temperature difference < Value</td></tr></table>Default: {0,0 0}</td></tr>
</table>

<table>
<tr><th>Input variable</th><th>Type</th><th>Description</th></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><td>State</td><td>Value</td><td>Description</td></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>
<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><td>State</td><td>Value</td><td>Description</td></tr><tr><td>0</td><td>0</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>0</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>nviLoopWind</td><td>SNVT_switch</td><td>Window contact information for looping when several controllers are present in the same room (refers to master / slave operation).

<table><tr><td>State</td><td>Value</td><td>Description</td></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 -1}</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.

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

<table>
<tr><th>Input variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nviOverRelay</td><td>SNVT_switch</td><td>Forcing of the electric heater relay. See chapter “4.3.14. Direct control of outputs” to see conditions.

<table><tr><th>State</th><th>Value</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>K contact close</td></tr><tr><td>1</td><td>100</td><td>K contact open</td></tr></table>The value field is not used.

Default : {0,0 0}</td></tr>
<tr><td>nviOverrideOcc</td><td>SNVT_occupancy</td><td>Occupancy forcing command, from a room operation unit or another control device (refer to nviOccManCmd also).

Range : OC_OCCUPIED, OC_UNOCCUPIED, OC_NUL
Default: OC_NUL</td></tr>
<tr><td>nviOverY1</td><td>SNVT_switch</td><td>Forcing of theY1 valve. See chapter “4.3.14. Direct control of outputs” to see conditions.

<table><tr><th>State</th><th>Value</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>Y1 output inactive</td></tr><tr><td>1</td><td>100</td><td>Y1 output active</td></tr></table>The value field is not used.

Default : {0,0 0}</td></tr>
<tr><td>nviOverY2</td><td>SNVT_switch</td><td>Forcing of theY2. See chapter “4.3.14. Direct control of outputs” to see conditions.

<table><tr><th>State</th><th>Value</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>Y2 output inactive</td></tr><tr><td>1</td><td>100</td><td>Y2 output active</td></tr></table>The value field is not used.

Default : {0,0 0}</td></tr>
<tr><td>nviOverY3</td><td>SNVT_lev_percent</td><td>Not supported by PCD7.L610 hardware
Unit: V Default: 0 Range: 0…10</td></tr>
<tr><td>nviOverY4</td><td>SNVT_lev_percent</td><td>Not supported by PCD7.L610 hardware
Unit: V Default: 0 Range: 0…10</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>
</table>

Input variable	Type	Description
nviSetptOffset	SNVT_temp_p	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
nviSpaceTemp	SNVT_temp_p	Value in °C used by the control loop and transmitted by the network. It is used in priority 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: -9.99..64.99

Output variable	Type	Description
nvoAlarm	SNVt_switch	Alarm input state – error flow control State | Value | Description 0 | 0 | Alarm Off, normal operation 1 | 100 | Alarm On, control loop disabled *Default: {0,0 0}*
nvoAnalogInput	SNVT_lev_percent	Not used
nvoAuxContact	SNVT_switch	State of the auxiliary contact State | Value | Description 0 | 0 | Closed contact (check polarity) 1 | 100 | Opened contact (check polarity) *Default: {0,0 -1}*
nvoChgOver	SNVT_switch	Change over switch state State | Value | Description 0 | 0 | Change over in warm mode 1 | 100 | Change over in cold mode *Default: {0,0 -1}*

<table>
<tr><th>Output variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nvoDewSensor</td><td>SNVT_switch</td><td>Dew point sensor value.

<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>Dew detection active</td></tr>
</table>
Default: {0,0 -1}</td></tr>
<tr><td>nvoDischAirTemp</td><td>SNVT_temp_p</td><td>Discharge air temperature.
Unit: °C Default: 327.67</td></tr>
<tr><td>nvoEffectOccup</td><td>SNVT_occupancy</td><td>Actual occupancy state of the controller.
Calculated from nviOccManCmd, nviOverrideOcc and nvoPresence.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>OC_OCCUPIED</td></tr>
<tr><td>1</td><td>OC_UNOCCUPIED</td></tr>
<tr><td>3</td><td>0C_STANDBY</td></tr>
</table>
Default: OC_OCCUPIED</td></tr>
<tr><td>nvoEffectSetpt</td><td>SNVT_temp_p</td><td>Value of the actual calculated temperature set point.
Unit: °C Default: 21</td></tr>
<tr><td>nvoElecCount</td><td>SNVT_time_hour</td><td>Electric heater operation time
Unit: hour Default: 0 Range: 0..65535</td></tr>
<tr><td>nvoEnergyHoldOff</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>Control loop disabled (but freeze protection remains active)</td></tr>
<tr><td>0xFF</td><td>0</td><td>Normal operation</td></tr>
</table>
Default: {0,0 0}</td></tr>
</table>

<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>nvoFanSpeedCmd</td><td>SNVT_switch</td><td>Fan speed command. See nviFanSpeedCmd.
Default: {0,0 -1}</td></tr>
<tr><td>nvoFlowControl</td><td>SNVT_switch</td><td>Flow rate detector switch state.

<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>Update nvoAlarm after 2 minutes</td></tr>
</table>
Default: {0,0 -1}</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>
<tr><td>9</td><td>HVAC_FAN_ONLY</td></tr>
</table>
Default: HVAC_OFF</td></tr>
<tr><td>nvoOADamper</td><td>SNVT_lev_percent</td><td>Not used.</td></tr>
<tr><td>nvoOccManCmd</td><td>SNVT_occupancy</td><td>Summary of occupation order of controller and network.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>-1</td><td>OC_NULL</td></tr>
<tr><td>0</td><td>OC_OCCUPIED</td></tr>
<tr><td>1</td><td>OC_UNOCCUPIED</td></tr>
<tr><td>3</td><td>0C_STANDBY</td></tr>
</table>
Default: OC_OCCUPIED</td></tr>
</table>

<table>
<tr><th>Output variable</th><th>Type</th><th>Description</th></tr>
<tr><td>nvoOutputReg1</td><td>SNVT_lev_percent</td><td>Process control output for Reg1
Unit: % Default: 0 Range: 0…100</td></tr>
<tr><td>nvoOutputReg2</td><td>SNVT_lev_percent</td><td>Process control output for Reg2
Unit: % Default: 0 Range: 0…100</td></tr>
<tr><td>nvoPresence</td><td>SNVT_occupancy</td><td>Presence detection input state (digital input on screw terminals or multisensor on RJ9 input)
<table><tr><td>Value</td><td>Description</td></tr><tr><td>-1</td><td>OC_NULL</td></tr><tr><td>0</td><td>OC_OCCUPIED</td></tr><tr><td>1</td><td>OC_UNOCCUPIED</td></tr><tr><td></td><td></td></tr></table>Default : OC_NUL</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>
<tr><td>nvoSpaceTemp</td><td>SNVT_temp_p</td><td>Value of the measured room temperature used by the control loop.
Unit: °C
Range: -9.99°C..64.99°C
Default: 327.67</td></tr>
<tr><td>nvoUnitStatus</td><td>SNVT_hVAC_status</td><td>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</td></tr>
</table>

<table>
<tr><th>Output variable</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>.fan_output (0): fan speed
Unit: % Range: 0…100

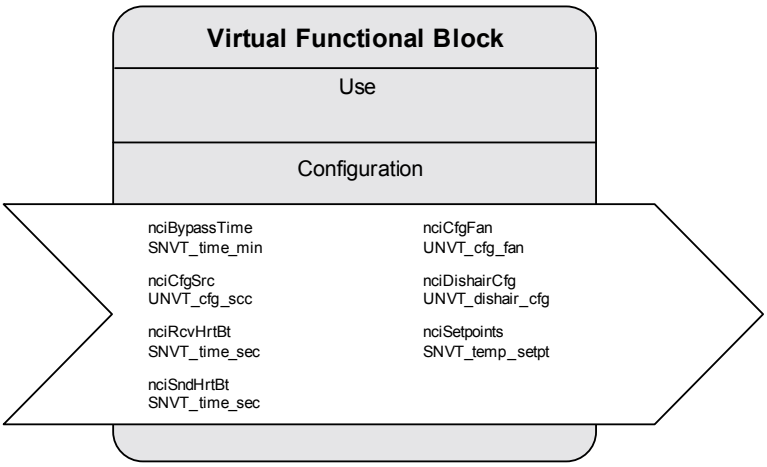
.in_alarm (0): error (0: no error)

Default : {HVAC_OFF,0,0,0,0,0,0}</td></tr>
<tr><td>nvoWindow</td><td>SNVT_switch</td><td>Window contact information used by the control loop.

<table>
<tr><th>State</th><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Window closed</td></tr>
<tr><td>1</td><td>100</td><td>Window open</td></tr>
</table>
Default : {0,0 -1}</td></tr>
</table>

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

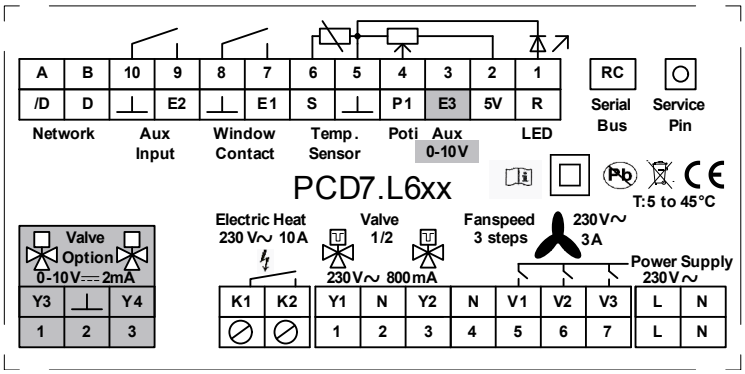
5.3. Virtual Functional Block



Configuration variable	Type	Description
nciBypassTime*	SCPTbypassTime SNVT_time_min	Same as ncBypassTime in the sccFanCoil functional block but in configuration network variable version.
nciCfgFan*	UCPTcfgFan UNVT_cfg_fan	Same as ncCfgFan in the sccFanCoil functional block but in configuration network variable version.
nciCfgSrc*	UCPTcfgScc UNVT_cfg_scc	Same as ncCfgSrc in the sccFanCoil functional block but in configuration network variable version.
nciDischairCfg*	UCPTdischairCfg UNVT_dishair_cfg	Same as ncDishairCfg in the sccFanCoil functional block but in configuration network variable version.
nciRcvHrtBt*	SCPTmaxRcvTime SNVT_time_sec	Not used.
nciSetpoints*	SCPTsetPnts SNVT_temp_setpt	Same as ncSetpoints in the sccFanCoil functional block but in configuration network variable version.
nciSndHrtBt*	SCPTmaxSendTime SNVT_time_sec	Same as ncSndHrtBt in the sccFanCoil functional block but in configuration network variable version.

6 Technical data

	Terminal	Description
Power supply	L, PE, N	230 VAC, + 10% / –15%, approx. 15 mA without current to TRIAC outputs Y1/Y2 and fan. An external fuse is required.
Ouputs		
Fan	N, V1, V2, V3	230 VAC, 3 A (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.
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...10 V	Input for Fan speed forcing
Voltage output	5 V	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 FT-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 26 580 30 00
Fax..... +41 26 580 34 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