Saia-Burgess Controls AG
Bahnhofstrasse 18 I 3280 Murten I Switzerland
T +41 26 672 72 72 I F +41 26 672 74 99 I www.saia-pcd.com

SOFTWARE DOCUMENTATION

Double room controller 230 VAC
PCD7.L615

Document version	Date	Updates	Soft version
1.0	12/05/2009	Initial version	1.00
1.1	31/07/2009	Add installation types 6 and 11. Add management of 2 channels radio accessories Correction window loop drawing. Correction lamp block drawing.	1.01
1.2	10/11/2009	Orhographic adaption	1.01
1.3	30/03/2011	Order numbers of Saia Burgess Controls	1.01
2.0	02/09/2013	New company name and company logo	1.01

PLAN

I. INTRODUCTION

I.1.General presentation

The PCD7.L615 are double loop HVAC controllers (230 VAC) communicating on the LonWorks® network. They allow to control climate room parameters in order to optimize users comfort. The The PCD7.L615 can manage 2 independent process loop control. Each controller can control indifferently warm or cold water devices, 2 or 4 tubes, with or without electric heater (optionnal). All infrared or radio accessories are fully compatible with the The PCD7.L615, directly connected to one of 2 RJ-9 input.

I.2.Interface

The device is a rectangular module supplying :

2 RJ-9 connectors, either for InfraRed Receiver (RIR) or room sensor.

1 10 points inputs connector:
- 3 points for 2 contact inputs (Main contact)
- 3 points for 2 mixed inputs (CTN or contact) (Sensor)
- 3 points for 2 mixed inputs (CTN or contact) (Aux contact)
- 1 point for 5V output

1 2 points connector for LON network

1 8 points connector for 230VAC outputs:
- 6 points for 4 valve ouputs 230 VAC
- 2 points for 2 relay outputs 230 VAC

1 3 points connector for 0-10V outputs:
- 3 points for 2 outputs 0-10V

2 2 points tie-point blocks for electric battery outputs
- 2 × 2points for 2 electric battery outputs

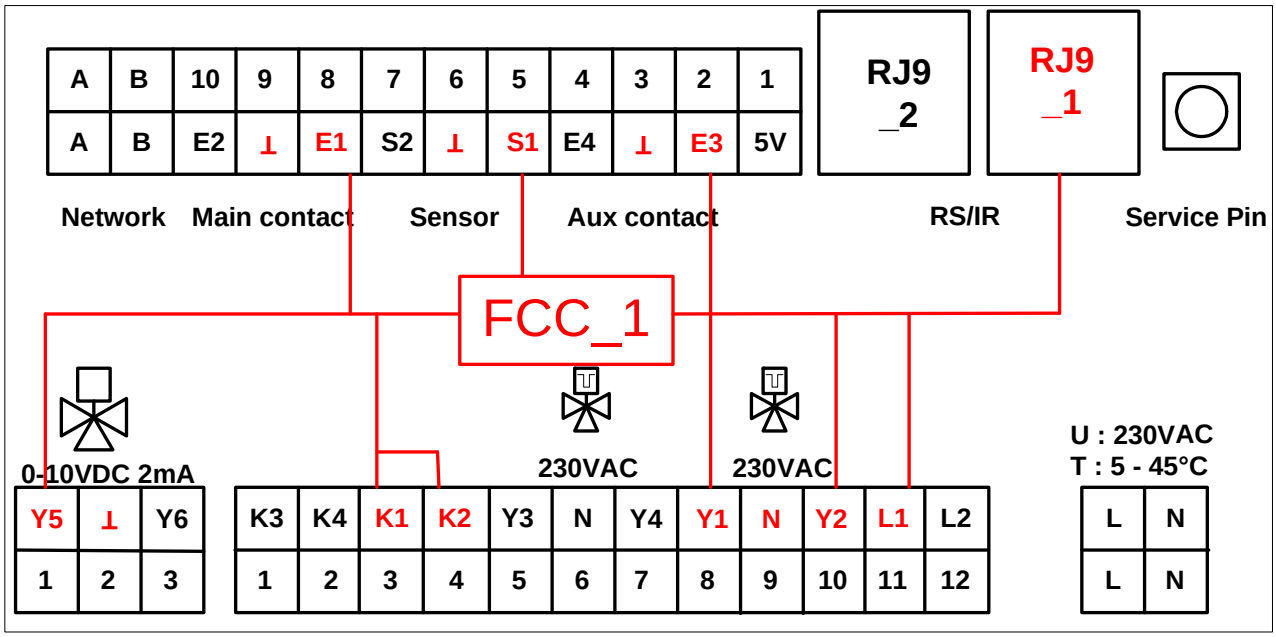
1 2 points power input connector (230 VAC)

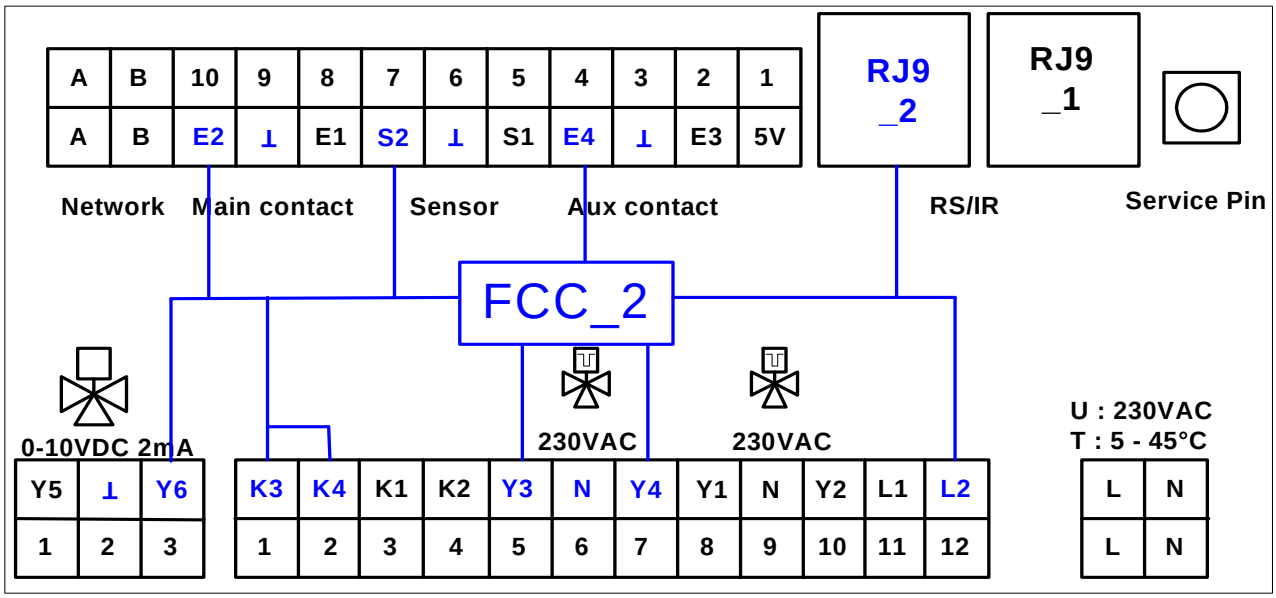
1 push button (service pin)

II. GENERAL FUNCTIONS

The PCD7.L615 controller contains:
2 independant "regulator" functional blocks, (sccFanCoil).
1 "node" functional block.
1 "config" functional block
4 "light" functional blocks

Each FCC "regulator" block has got dedicated inputs and outputs, and controls its ouputs according to its own regulation loop.





III.CONFIG OBJECT

The config object provides all the network variables and configuration properties about the general configuration of the device. These configurations are applied for both regulation loops.

III.1. INPUTS / OUTPUTS SETTINGS

III.1.1. Inputs settings

Input name	Terminal	Contact	Temp sensor
Main FCC2	E2	X	
Main FCC1	E1	X	
Sensor FCC2	S2	X	X
Sensor FCC1	S1	X	X
Auxiliary FCC2	E4	X	
Auxiliary FCC1	E3	X	

Function	Code for nciInputCfg	Updated variable	E2	E1	S2	S1	E4	E3
Not Used	0xFF (255)							
Window	0	nvoEnergyHold Off						
Occupancy	1	nvoPresence						
Dewpoint	2	-						
Change Over	3	-						
Auxiliary contact (state report)	4	nvoAuxContact						
Flow controller	5	-						
Space or return temperature	10	nvoSpaceTemp						
Default value			0	0	10	10	0xFF	0xFF

III.1.2. Outputs settings

Output name	Terminal
0-10V FCC1	Y5
0-10V FCC2	Y6
Electric relay (EB) FCC1	K1-K2
Electric relay (EB) FCC2	K3-K4
230V Triac 1 - FCC1	Y1
230V Triac 2 - FCC1	Y2
230V Triac 1 - FCC2	Y3
230V Triac 2 - FCC2	Y4
230V Relay – FCC1	L1
230V Relay – FCC2	L2

Fonctions	Code for ncOutputCfg	Y5	Y6	K1-K2	K3-K4	Y1	Y2	Y3	Y4	L1	L2
Not used	0xFF (255)										
0-10V	1										
Electric battery relay	2										
Regulation output Reg1	3										
Regulation output Reg2	4										
230V Relay	5										
Default value		**1**	**1**	**2**	**2**	**3**	**4**	**3**	**4**	**7**	**7**

Note :
The 230 V triac outputs manage **ONLY** thermic valves (no “3 points” valves).

IV. REGULATOR OBJECT

Each FCC « régulateur » block is designed to manage the following functions:

RJ-9 input:
1 RJ-9 input to connect a room sensor, multisensor or infrared remote to receive the following informations:
- setpoint offset
- occupancy / unoccupancy
- space temperature
- presence detection
- fan speed commands (for variable speed (0-10V) ventilation)
- lights ON/OFF commands

Terminal inputs:
3 configurable inputs:
- window contact
- change over contact
- présence detection contact
- dewpoint contact
- flowcontrol contact
- space or return temperature sensor

(See also the allowed settings in chapter « Inputs configuration »)

Terminal outputs:
- 2 × 230 V triacs for 2 thermic valves.
- 1 × 230 V relay for electric battery max 1,25 kW.
- 1 × 0-10 V output for controlling of 0-10 V damper or 0-10 V variable speed fan or 0-10 V valve output.
- 1 × 230 V relay output for power supplying of variable speed fan, or controlling of ON/OFF lamp, or managed by the network.

Main functions:
- Space temperature regulation (PI)
- Flowcontrol security management
- Postventilation management
- Antifreeze management
- Occupancy scenarii management
- Lights commands management

IV.1. PROCESS CONTROL DETAILS

IV.1.1. Setpoint computation

The controller computes a cool setpoint and a heat setpoint according to the following algorithm:
a) Computation of a setpoint offset: *offset1*
If nviSetpoint is valid: *offset1 = nvisetpoint – (nciSetpoints .unoccupied_cool + nciSetpoints. unoccupied_heat)/2*
else *offset1 = 0*

b) Computation of the user offset sent by the room sensor: *offset2*

Unoccupied mode :
- *Heat setpoint = nciSetpoints. unoccupied_heat*
- *Cool setpoint = nciSetpoints .unoccupied_cool*

No offset is applied on the setpoint

Standby mode
- *Heat setpoint = nciSetpoints. standby_heat + offset1 + offset2*
- *Cool setpoint = nciSetpoints .standby_cool + offset1 + offset2*

Occupied or Bypass mode
- consigne chaude = nciSetpoints. occupied_heat + offset1 + offset2
- consigne froide = nciSetpoints .occupied_cool + offset1 + offset2

The HVAC_MRNG_WRMUP mode is not managed.

IV.1.2. Computation of the final regulation command

Definitions and computation rules
PctOuv(t) is the final percentage applied on the regulation actuator (valve or electric battery) PctOuv(t)= Proportional(t) + Intégral(t) The integral value is limited at 100 % Variables are evaluated and updated every 10 seconds. Computation of the temperature difference HVAC_HEAT : Difference(t) = Heat setpoint – Space temperature HVAC_COOL : Difference(t) = Space temperature – Cool setpoint According to these settings, the opening percentage of the valve or the activation level of the electric battery will be computed.

Note:
The modification of a configuration parameter is not taken in account immediately. The new value will be applied during the next regulation computation. The regulation time cycle is 10 seconds.

IV.1.3. Valve outputs

In order to act in priority on the valves rather than on the ventilation, the ncValveCoeff parameter is used to apply a multiplicative ratio on the regulation result.
ncValveCoeff should be set between 0 and 250.

- 0 = no action on the valve
- 100 = same action on the valve as on the ventilation
- 200 = valve is opened twice the regulation result

The result applied on the valve is limited to 100 %

IV.1.4. Time considerations

The control loop is exectuted every 10 seconds.
However, to garantee a fast response time on critical actions, this computation is done immediately in the following cases:

- Modification of the fan speed (nviFanSpeedCmd or user command from remote controller/room sensor)
- Modification of the window state (nvoWindow)

IV.1.5. Antifreeze mode

The antifreeze mode has priority on any other mode and is always active.
If spacetemp ← Antifreeze limit, then the fanspeed is set to its maximal value, heat valve and/or electric battery are forced to 100%.
When the antifreeze actions are enabled, nvoHeatCool = HVAC_EMERG_HEAT.

IV.1.6. Window/door opening detection

The nviEnergyHoldOff variable and the window contact (nvoWindow) are used to dertermine if a window is opened.
In this case, the regulation is disabled (valves closed, fan stopped and electric battery stopped) as long as the window remains opened.

In this mode, the regulator doesn't allow fanspeed forcing, doesn't take into account the roomsensor's commands and stops (if configured), the small speed fan forcing into the dead zone.
When using a bidirectional RS-LCD, an alarm is displayed on the screen.

IV.1.7. Electric heater fan control

If the network fanspeed command (nviFanSpeedCmd) sets the fan OFF, the electric heater command will be forced to zero too.
For every fan stop command, the small speed should be maintained during a configurable time (nciCfgFcc.fanOffDelay).

If flow control function is enabled, a lack of flow will result into a fan stop command.

IV.1.1. Room temperature measurement

It can be provided by several sensors:
- The analogic temperature sensor connected to the controller.
- The infrared/radio remote controller or room sensor, via the RJ-9 link.
- The nviSpaceTemp network variable (from another device/BMS).

The control manages the priority from these temperatures on the following way:
1/ Network variable nviSpaceTemp if it is valid (→-10 °C and←65 °C)
2/ Analogic sensor or room sensor depending on nciCfgFcc.SensorSelect

If temperature source is from RJ-9, update frequency is checked. With no temperature update within 1 hour, the variable nvoSpaceTemp becomes TEMP_INVALIDE and the controller disables the regulation.

IV.1.2. 0-10V output computation

The 0-10V output can be managed:
As a 0-10V fan, according to the regulation command *(ncCfgFan.mode = 1 or 2)*
The output is follows the regulation result on the 0-10 V range. A minimum value can be parametered.

According to the occupancy state *(ncCfgFan.mode = 3)*
Depending on the occupancy (Occupied, Unoccupied, Standby), the 0-10 V output is set to a configurable value (occupied: ncCfgFan.level3, standby : ncCfgFan.level2, unoccupied: ncCfgFan.level1).

According to a network variable *(ncCfgSrc.auxCmdType = 5)*
Output is controlled by network variable (nviAuxCmd), with ability to set a minimum value

If the 0-10V output is used as a fan command, it is possible to disable the fan in heat and/or cool using the configuration parameter nciCfgFan.cfg:
0 normal
1 no fan
2 no fan in cooling
3 no fan in heating
Fan is stopped in dead zone. However, it can be forced with ncCfgFan.override:
0 no forcing
1 mini V1 in occupied and standby modes
2 mini V1 in occupied and standby modes but stop allowed if requested by the user (RJ-9 command).
3 mini V1 every modes
4 same as 2, with 5 minutes restarts every 2 hours in other modes.
The values level1, level2 and level3 of nciCfgFan correspond to the 3 speeds V1, V2 and V3 when forcing requests come from the RJ-9.

By configuration, fan can be controlled as a « 3 speeds fan» in automatic mode, based on the 3 levels V1, V2, V3 defined above.

It is also possible to define a minimum fan value. In this case, regulation works between this minimum and V3 : example :
- mini is configured to 20 and V3 to 100.
- If 0 % requested → output 0
- If 20 % requested → output 36
- If 100 % → output 100

It is possible to invert the command signal if ncFunctionCfg.fancontrol=1. Example 80 % → 2 V.

IV.1.3. Dewpoint management

If an input configured into dew point is active:
→ the ventilation follows the regulation signal or the forcing fan speed parameters.
→ the regulation cooling command is forced to 0 % (value displayed into the nvoUnitStatus)

IV.1.4. Contacts actions on the regulation

Window	**ncFunctionCfg.window**	**nvoMainContact ou nvoAuxContact**	**Effect**
Contact opened	0	{0 0}	Regulation active
Contact opened	1	{1 100}	Regulation stopped
Contact closed	0	{1 100}	Regulation stopped
Contact closed	1	{0 0}	Regulation active
Dew	**ncFunctionCfg.dew**	**nvoMainContact ou nvoAuxContact**	**Effet**
Contact opened	0	{0 0}	No effect
Contact opened	1	{1 100}	Heat regulation only Cool forced to 0
Contact closed	0	{1 100}	Heat regulation only Cool forced to 0
Contact closed	1	{0 0}	No effect
Change Over	**ncFunctionCfg.chgover**	**nvoMainContact ou nvoAuxContact**	**Effet**
Contact opened	0	{0 0}	Heat mode
Contact opened	1	{1 100}	Cool mode
Contact closed	0	{1 100}	Cool mode
Contact closed	1	{0 0}	Heat mode
Auxiliaire	**ncFunctionCfg.auxiliary**	**nvoMainContact ou nvoAuxContact**	**Effet**
Contact opened	0	{0 0}	No effect (info)
Contact opened	1	{1 100}	No effect (info)
Contact closed	0	{1 100}	No effect (info)
Contact closed	1	{0 0}	No effect (info)
FlowControl	**ncFunctionCfg.flowcontrol**	**nvoMainContact ou nvoAuxContact**	
Contact opened	0	{0 0}	Elec heater disabled
Contact opened	1	{1 100}	Elec heater enabled
Contact closed	0	{1 100}	Elec heater enabled
Contact closed	1	{0 0}	Elec heater disabled

IV.1.5. Network variables heartbeat

It is possible to configure a HeartBeat (*nciSndHrtBt*) to propagate the following network variables:
nvoEffectOccup
nvoHeatCool
nvoWindow
nvoMainContact
nvoAuxContact

IV.1.6. Electric heater limitation/load sheeding

It is possible to limit the output command of the electric battery.

→ If nviEconEnable.state=0 → no limitation
→ If nviEconEnable.state=1 → electric battery output limited to nviEconEnable.value
→ If nviEconEnable.state=0xFF (auto) : load sheeding if the temperature difference is above nviEconEnable.value (in tenth of a second)

nviEconEnable.state	nviEconEnable.value	Limitation/Load shedding
0	n/a	No limitation
1	20	Electric battery limited to 20%
1	0	Elecctric battery limited to 0%
0xFF (-1)	10	Electric battery stops if nvoEffectSetpt – nvoSpaceTemp ← 1 °C
0xFF	0	Electric battery stops if nvoEffectSetpt = nvoSpaceTemp
0xFF	200	Electric battery stops if nvoEffectSetpt – nvoSpaceTemp ← 20 °C

The electric heater limitation/load shedding function (nviEconEnable) applies on the two regulation loops (FCC1 and FCC2)

IV.1.7. Room sensors configuration

The PCD7.L615 is compatible with the following numeric (RJ-9) room sensors:
RS-DLx, RS-LCDx

Parameters to apply for a numeric room sensor:

- nciCfgFcc.roomModuleType = 0
- nciCfgFcc.sensorSelect = 0 → Use the analogic temperature sensor Sx.
- nciCfgFcc.sensorSelect = 1 → Use the numeric temperature sensor from RJ-9.
- nciCfgFcc.roomModuleDisplay → To set the displayed information on the RS-LCD screen.

The room sensor connected on the RJ-9_1 always commands the FCC1 regulation loop.
The room sensor connected on the RJ-9_2 always commands the FCC2 regulation loop.

Use of radiofrequency accessories :
The commands received by a RFR-K connected on the RJ-9_1 are applied to the FCC1 regulation loop.
The commands received by a RFR-K connected on the RJ-9_2 are applied to the FCC2 regulation loop.

A unique RFR-D can be equally connected on the RJ-9_1 or RJ-9_2. Only one is required to command both regulation loops.
The commands coming from an accessory (TCND-R, WMSPB-8DI) configured into zone 1 (canal 1) are applied to the FCC1 regulation loop.
The commands coming from an accessory (TCND-R, WMSPB-8DI) configured into zone 2 (canal 2) are applied to the FCC2 regulation loop.

IV.1.8. L1 and L2 230V relay outputs

The device has got 2 × 230 V relay outputs L1 and L2.
Each regulation block controls **one** of these 230 V output.

These outputs can be controlled:

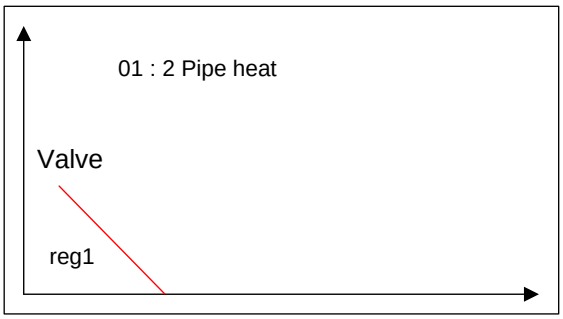
- Individualy through the LON network: nviCmdRelay variable
- By the fan speed commands from the regulation blocks: the relay can be configured as 230 V power supply for variable speed fan. Its activation depends of the fan speed command. The relay remains active during the postventilation period.

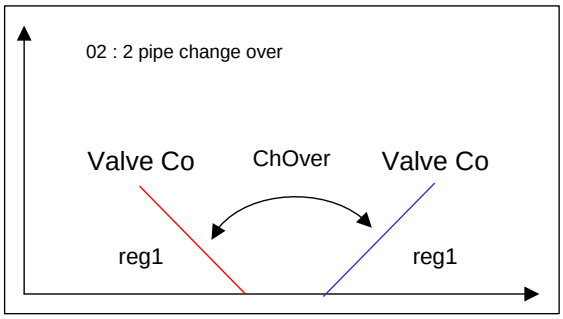
In both cases, the FCC1 functionnal block commande the L1 output, the FCC2 functionnal block commande the L2 output.

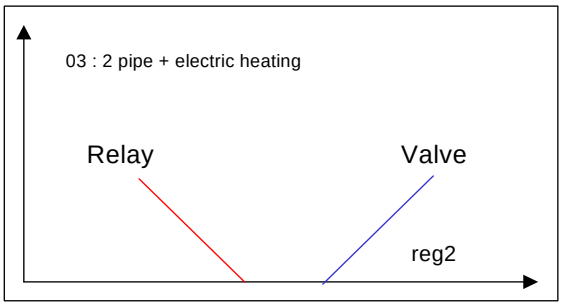
- By the light commands from the regulation blocks: with a configuration, each relay is associated to a lamp functional block.
  If the lamp is switched ON: relay closed.
  If the lamp is switched OFF: relay opened.

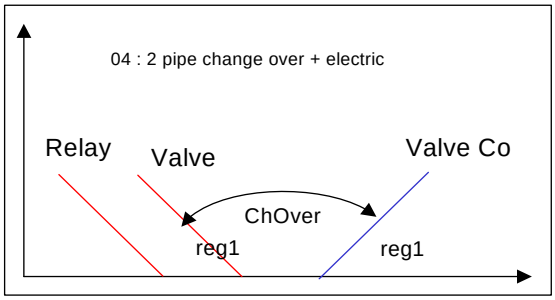
IV.1.9. Configurations details

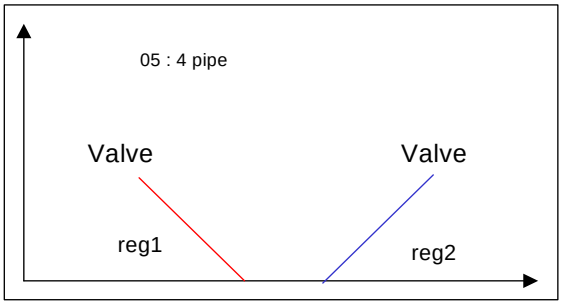
The installation type should be parametered in nciCfgSrc.type
The supported installations types are:

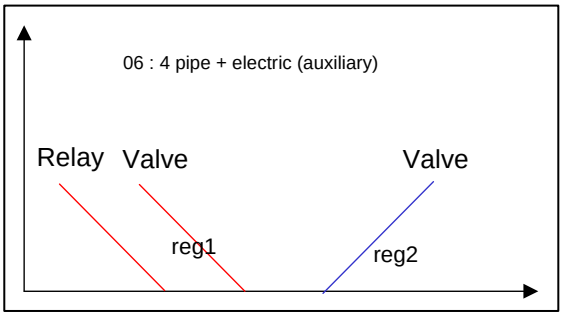


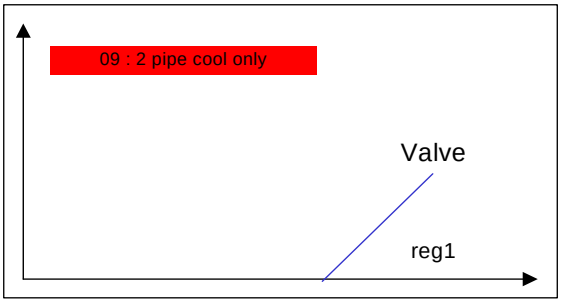


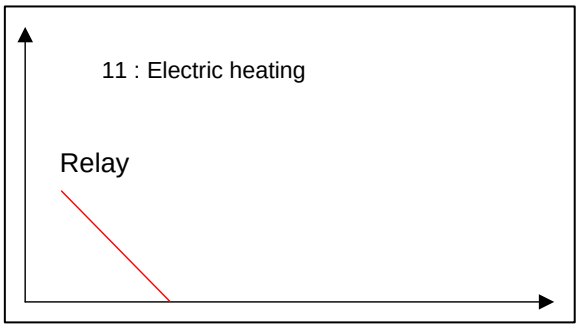












type	description	Change Over reg1	Heat valve reg1	Cool valve reg2	Electric heater relay
01	2 pipes heat				
02	2 pipes change over				
03	2 pipes + electric heater				
04	2 pipes change over + electric heater	primary			secondary
05	4 pipes				
09	2 pipes cool				

IV.1.10. Occupancy modes management

The effective room occupancy is the result of the BMS occupancy (3 states allowed) and of the room occupancy command from the presence detection or from the roomsensor/network forcing (2 states allowed).
This remains into 6 occupancy scenarios. For each scenario, sereral actions on lights and HVAC are done according to the parmeters
By default, les régimes d'occupation sont configurés de la manière suivante:

BMSOccupancy/RoomOccupancy	Room Occupied	Room Unoccupied
BMS Occupied	Lights ON nvoEffectOccup =OC_OCCUPIED	Lights OFF nvoEffectOccup =OC_UNOCCUPIED
BMS Unoccupied	Lights ON nvoEffectOccup =OC_OCCUPIED	Lights OFF nvoEffectOccup =OC_UNOCCUPIED
BMS Standby	Lights ON nvoEffectOccup =OC_OCCUPIED	Lights OFF nvoEffectOccup =OC_STANDBY

The effective occupancy state used by the HVAC regulation appears in the nvoEffectOccup.

The BMS occupancy state appears in the **nviOccManCmd**. This variables allows 3 values : OC_OCCUPIED (0), OC_STANDBY(3) and OC_UNOCCUPIED (1). Any other value is considered as OC_OCCUPIED.

If **ncRegulOcc.GtbOcc** = 1(or 3) and nviOccManCmd goes OC_OCCUPIED, the regulator goes into BMSOccupancy=Occupied and into RoomOccupancy=Occupied (equivalent to a remote forcing command)
If **ncRegulOcc.GtbOcc** = 2(or 3) and nviOccManCmd goes OC_UNOCCUPIED, the regulator goes into BMSOccupancy=Unoccupied and into RoomOccupancy=Unoccupied (equivalent to a remote forcing command)

The RoomOccupancy state is the computation of:

- The presence detection sensors (RJ-9 or contact).
- The action of the user on the remote controller or on nviOverrideOcc.

When the user presses the Occupied or Unoccupied buttons of the remote controller, the room goes into the corresponding occupancy state.
On each presence detection, the room occupancy timer restarts at ToPresence.
When no presence detection occurs during ToPresence, the room goes unoccupied.

Occupied button on remote controller: Room goes occupied, if ncRegulOcc.TcndOcc is not 0 then the room remains occupied during ToPresence, and goes back unoccupied after this delay.
Warning: when BMS occupancy is unoccupied, the room remains occupied during ToPresence whithout considering the ncRegulOcc.TcndOcc value.

If ncRegulOcc.TcndOcc=0, the room remains occupied until the end of the presence detection or a push on the unoccupied button of the remote control.

Unoccupied button on remote controller: Room goes unoccupied, if ncRegulOcc.TcndInocc is not 0 then the presence detection is disabled during ncRegulOcc.TcndInocc (in sec). If ncRegulOcc.TcndInocc=0 then the presence detection is disabled as long as the room is occupied (presence detected).

When nviOverrideOcc goes occupied or unoccupied, the regulator behaviour is the same as a occupied/unoccupied push on the remote controller.

IV.1.11. Action when the occupancy state changes

When the occupancy state changes, a command is sent to the lighting devices and to the sunblinds via nvoCmdLum and nvoCmdStores. This command cancels any forcing from the remote controller or BMS and allows immediate control of the lighting devices and sunblinds as well as the start of the automatic light control according to cpRegulOcc.

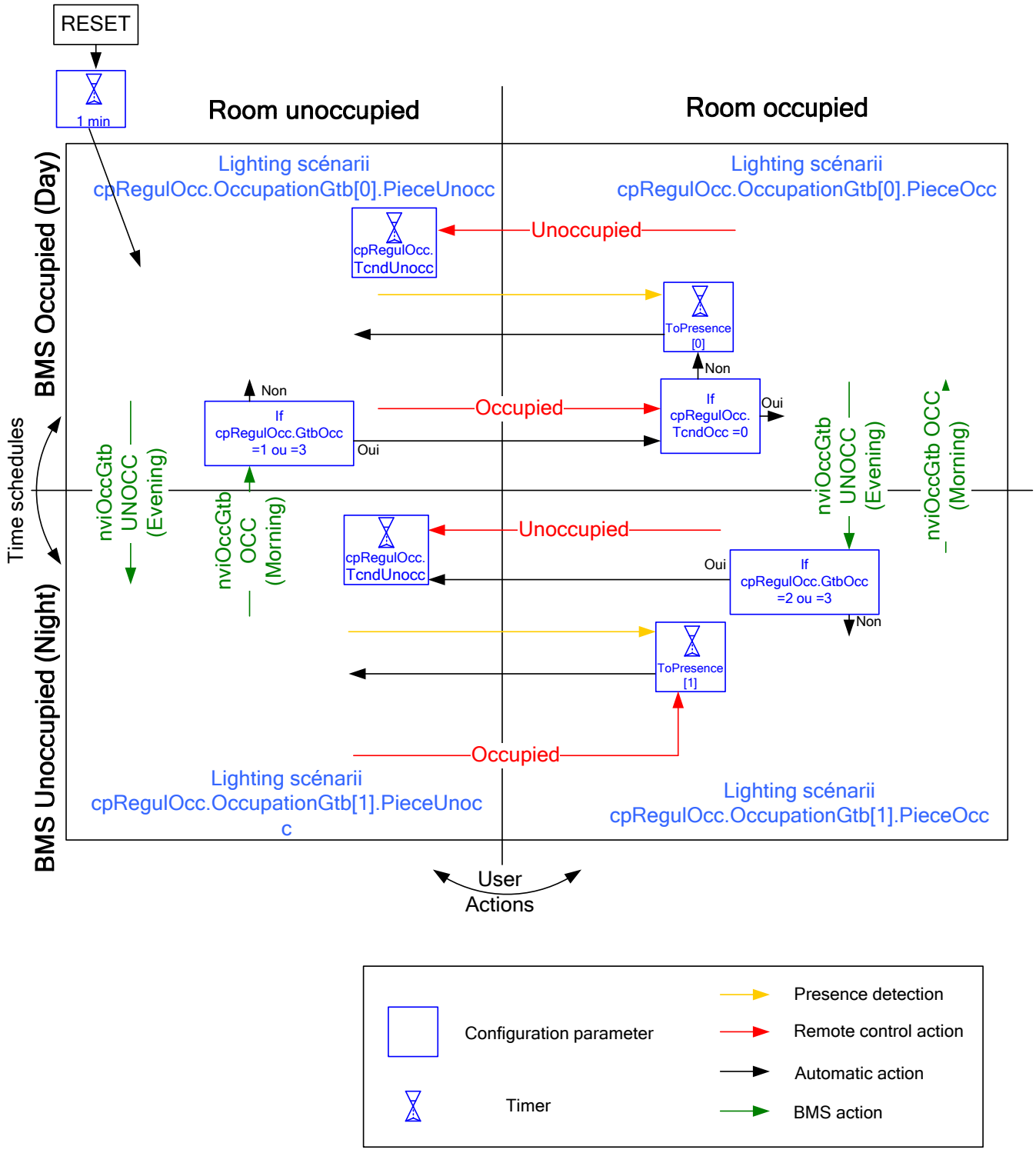
According to the occupancy state, the object uses the relevant parameter

Régime d'occupation	Paramètre utilisé
BMS Occupied, room Occupied cpRegulOcc.OccupationGtb[0].PieceOcc	BMS Occupied, room Unoccupied cpRegulOcc.OccupationGtb[0].PieceUnocc
BMS Unoccupied, room Occupied cpRegulOcc.OccupationGtb[1].PieceOcc	BMS Unoccupied, room Unoccupied cpRegulOcc.OccupationGtb[1].PieceUnocc
BMS standby, room Occupied cpRegulOcc.OccupationGtb[2].PieceOcc	BMS standby, room Unoccupied cpRegulOcc.OccupationGtb[2].PieceUnocc
BMS Occupied, room Occupied cpRegulOcc.OccupationGtb[0].PieceOcc	BMS Occupied, room Unoccupied cpRegulOcc.OccupationGtb[0].PieceUnocc
BMS Unoccupied, room Occupied cpRegulOcc.OccupationGtb[1].PieceOcc	BMS Unoccupied, room Unoccupied cpRegulOcc.OccupationGtb[1].PieceUnocc
BMS standby, room Occupied cpRegulOcc.OccupationGtb[2].PieceOcc	BMS standby, room Unoccupied cpRegulOcc.OccupationGtb[2].PieceUnocc

The variable nvoOccEffect takes the la value contained in OccVal.
CmdLumG1 = 1: set the « window side » light intensity to LumGrada1 (x10%)
Or = value around the stable loop process if LumGrada1 =11
CmdLumG1 = 0: does not control the « window side » lights.
CmdLumG2: same as CmdLumG1 for the « corridor side » lights.

IV.1.12. Links between occupancy states :

BMS in standby mode not described below (= BMS« occupied »).



V. LIGHT OBJECT

V.1. EXTENSION DEVICES

The Light functionnal object commands a ON/OFF lamp which sould be connected to the following extension devices:
PCD7.L620
PCD7.L621

The lights commands are only the result of user commands (remote control) or presence detections ➔ lighting regulation with ambient lighting measurement is NOT supported.

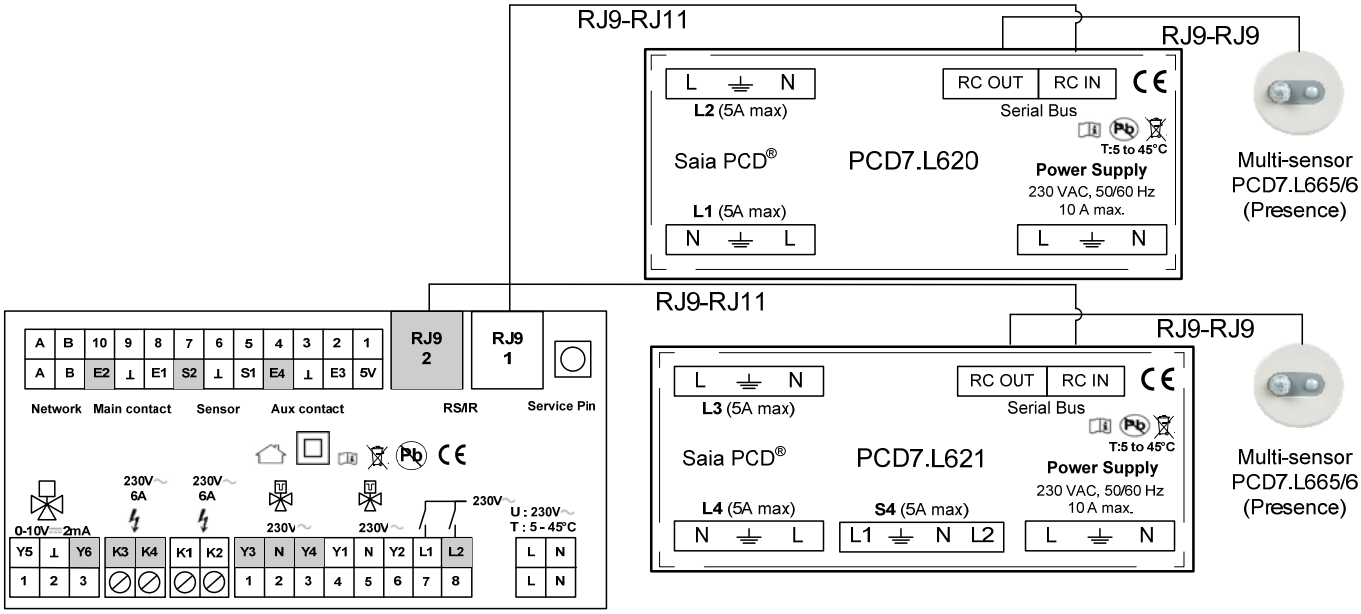
On each sccFanCoil funcionnal block, the nvoCmdLum variable transmits light commands coming from its associated RJ-9 input.

V.2. LIGHTING MANAGEMENT

The PCD7.L615 device can manage 4 ON/OFF lights (0 or 100%).

V.2.1. Default behaviour (no bindings)

The factory settings are the following:
Light1 et Light2 functionnal blocks command the lights L1 and L2 of an PCD7.L620 connected on the RJ-9_1 using the nvoCmdLum of FCC1 block.
Light1 et Light2 functionnal blocks command the lights L3 and L4 of an PCD7.L621 connected on the RJ-9_2 using the nvoCmdLum of FCC2 block.



V.2.2. Specific behaviour

To modify the physical outputs/Light functionnal blocks affectation, the cpLumCfg.Connection (1 byte) parameter should be used:
Bit8 : RJ-9 connector number of the light to be set.
0 = command sent on RJ-9_2, 1 = command sent on RJ-9_1
Bit7 : not used
Bit6 to Bit1: extension device type and output light number.

VI. NODE OBJECT

The node object provides the required LonMark network variables for a LonWorks network use of the device.

VII. MASTER SLAVE CONFIGURATION

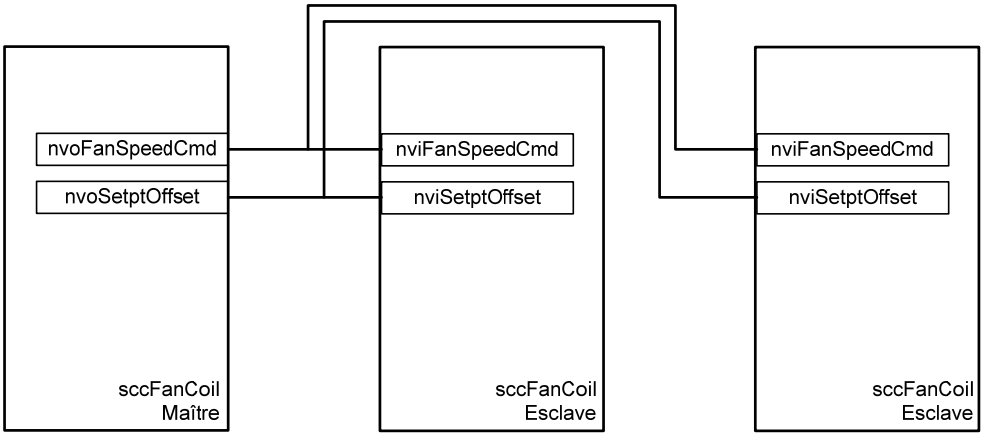
When installing several controllers in the same room, they should have the same behaviour. To do this, one controller should be declared as master, sending by binding some information to the other controllers.

The following natwork variables should be transmited to the slaves:
nvoFanSpeedCmd
nvoSetptOffset
nvoEnergyHoldOff
nvoHeatCool
nvoAuxContact

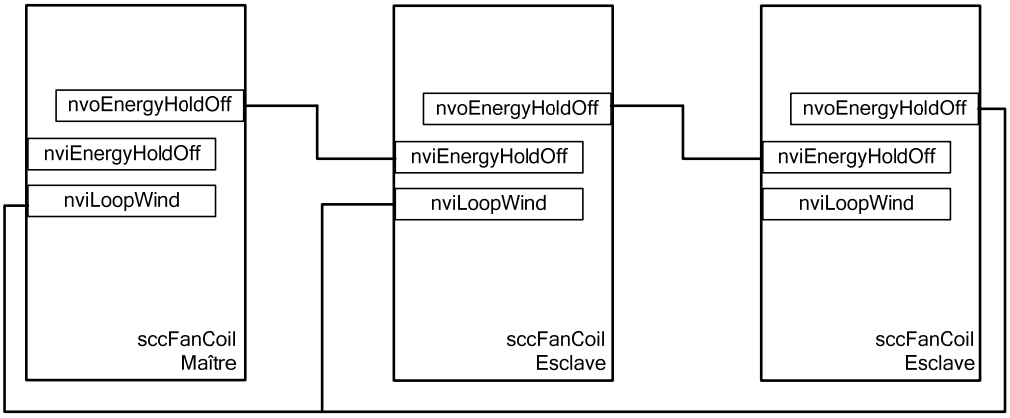
The (nvi) variable which will receive the nvoAuxContact information from the master is nviAuxCmd. By configuration, the nature of the information should be parametered as one of the following:
1 = Presence contact
2 = Dewpoint contact
3 = Change Over contact
4 = FlowControl contact

In a master/slave mode, one of the devices (and its regulation block) is said “master”. The slave can be either the other regulation block from same controller or either another regulation block from another controller.
The master/slave links to be done depend of the number of room sensors in the room.

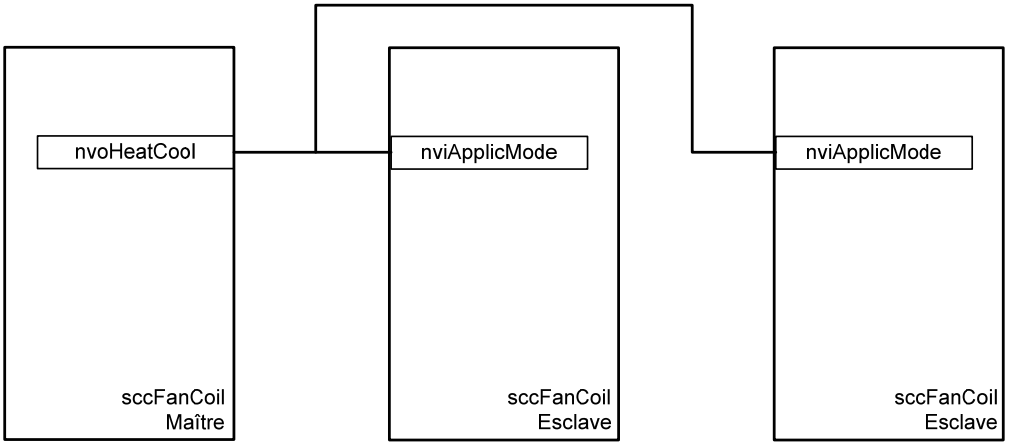
Fan speed command and setpoint offset master/slave links:



Window master/slave links:

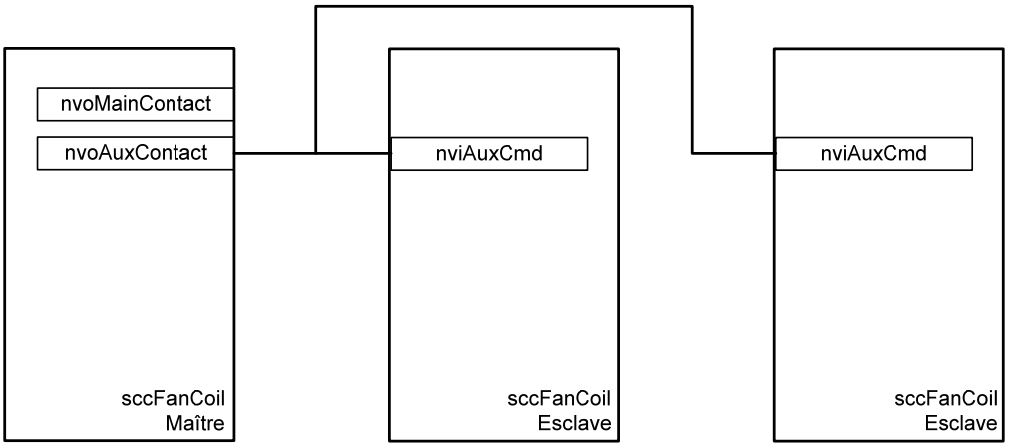


Application mode master/slave links:



Auxilliairy contact master/slave links :

Warning : the nature of the information coming in nviAuxCmd must be parametered with nciCfgSrc.auxCmdType

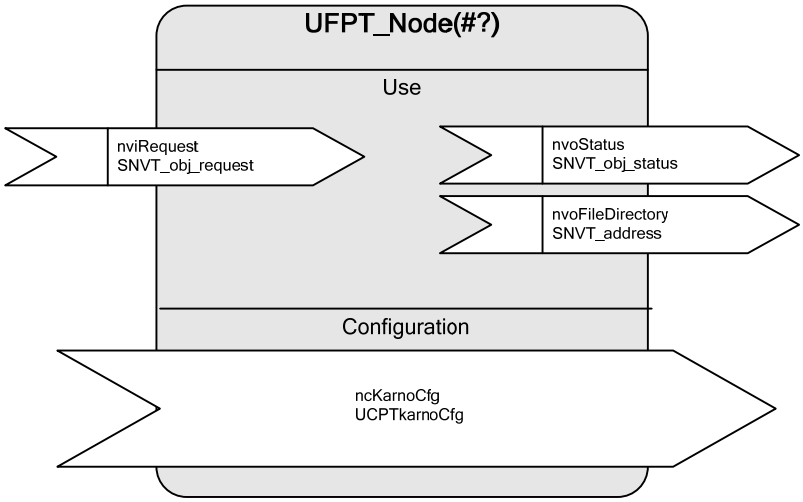


VIII. FUNCTIONNAL COMPARISON WITH PCD7.L615

- HVAC regulation functions are like SRC with variable speed fan control instead of 3 speeds fan control.
- Adding of a flow control function (physical input and network variables, positive security.
- No counting management
- No links with KarnoWeb / ZcDisplay
- No air quality management
- No air damper management
- No dischair temperature limitation

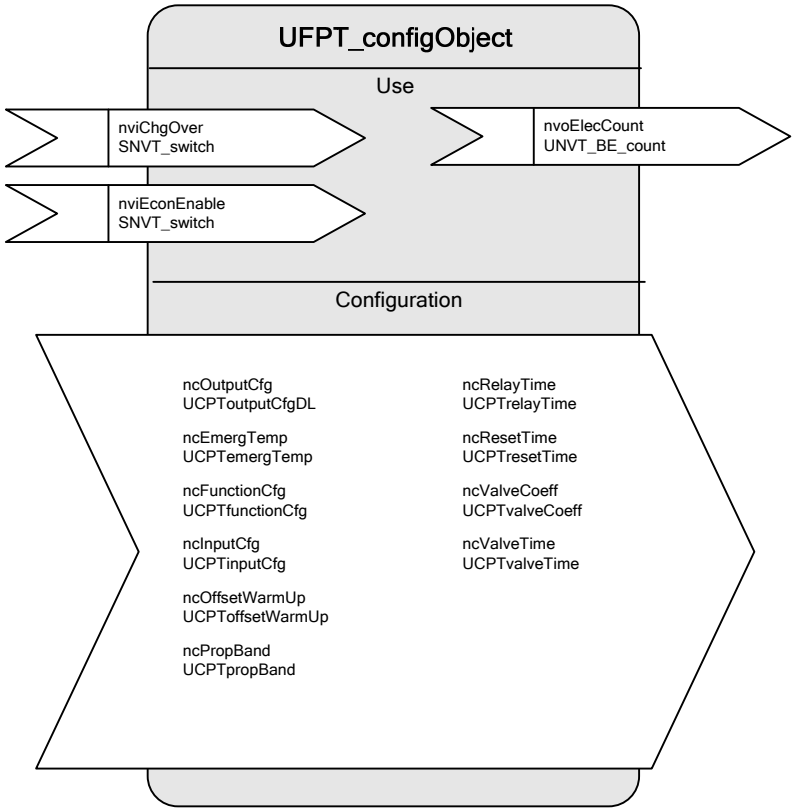
IX. LONMARK PROFILS DESCRIPTION

IX.1. NODE OBJECT



Config Prop	Type	Description
ncKarnoCfg	UCPTkarnoCfg	**Internal Karno configurations**
Variable Entrée		
nviRequest	SNVT_obj_request	**Node status request**. Only node object requests (#0) of type RQ_NORMAL, RQ_UPDATE_STATUS et RQ_REPORT_MASK are allowed.
Variable Sortie		
nvoStatus	SNVT_obj_status	**Node status.** nvoStatus updated in response to nviRequest and when node reset.
nvoFileDirectory	SNVT_address	**Variables structure** (mandatory)

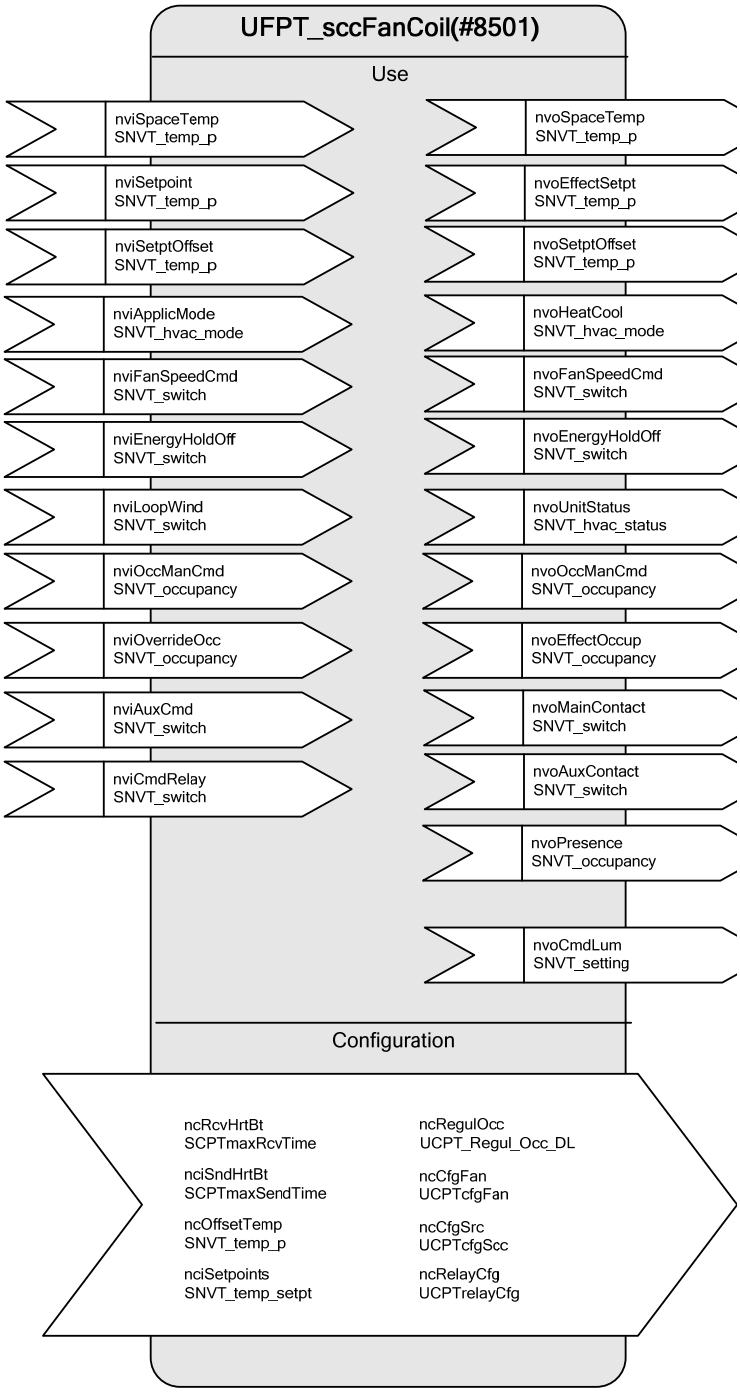
IX.2. CONFIG OBJECT



Config Prop	Type	Description
ncOutputCfg	UCPTOutputCfgDL { Y5 Y6 K1K2 K3K4 Y1 Y2 Y3 Y4 L1 L2 manuf11 manuf12 }	**Output configurations** (See codes **III.1.2** page **6**) .**Y5** : Y5 output configuration .**Y6** : Y6 output configuration .**K1K2** : K1K2 output configuration .**K3K4** : K3K4 output configuration .**Y1** : Y1 output configuration .**Y2** : Y2 output configuration .**Y3** : Y3 output configuration .**Y4** : Y4 output configuration .**L1** : L1 output configuration .**L2** : L2 output configuration .**manuf11** : reserved .**manuf12** : reserved By default : { 1,1,2,2,3,4,3,4,5,5,255,255}
ncEmergTemp	UCPTemergTemp	**Anti freeze temperature configuration** By default = 8°C Range: 0 to 20°C
ncFunctionCfg	UCPTfunctionCfg { window chgover dew presence heatvalve coolvalve auxiliary flowcontrol fancontrol manuf2 manuf3 }	**Inputs/outputs polarity configuration (NO/NF)** .**window** (0 = NO / 1 = NF) .**chgover** (0 = open for heat / 1 = closed for heat) .**dew** (0 = NO / 1 = NF) .**presence** (0 = opened for occupied / 1 = closed for occupied) .**heatvalve** (0 = NF (valve closed) / 1 = NO (valve open)) .**coolvalve** (0 = NF (valve closed) / 1 = NO (valve open)) .**auxiliary** (0 = NO / 1 = NF) .**flowcontrol** (0 = NO / 1 = NF) .**fancontrol** (0 direct / 1 reverse) .**manuf2** : reserved .**manuf3** : reserved By default {0,0,0,0,0,0,0,0,0,0}
ncInputCfg	UCPTinputCfgDL { E2 E1 S2 S1 E4 E3 Manuf7 }	**Inputs configuration** *(See codes **III.1.1** page **5**)* .**E2** : E2 input confiugration .**E1** : E1 input confiugration .**S2** : S2 input confiugration .**S1** : S1 input confiugration .**E4** : E4 input confiugration .**E3** : E3 input confiugration .**manuf7** : reserved By default : {0, 0, 10, 10, 255, 255, 255}
ncOffsetWarmUp	UCPToffsetWarmUp	**Setpoint Offset for HVAC_MRNG_WRMUP mode** By default : 0°C
ncPropBand	UCPTpropBand	**Proportionnal band used by HVAC regulation in °C** mini 2 °C – maxi 20 °C By default : 5 °C

ncResetTime	UCPTresetTime	**Integral time (in sec)** 0 = integral disabled Minimum value 60 seconds By default ncResetTime = 600 seconds
ncRelayTime	UCPTrelayTime	**Cycle time 230V relay K in seconds** mini 100s - maxi 250s By default 240s
ncValveTime	UCPTvalveTime	**Cycle time for 230V thermic valve outputs** By default 20 s – maxi 250 s
ncValveCoeff	UCPTvalveCoeff	**Valve output coefficient** Range: 0 to 250 By default : 100
Input variables		
nviChgOver	SNVT_switch	**Change Over Mode information** ({0,0}= Heat {100,1} =Cool Stored in EEPROM (limited write number) By default = {0,0}
nviEconEnable	SNVT_switch	**Electric heater output limitation** See load sheeding chapter IV.1.6 By default : {0,0}
Output variables		
nvoElecCount	UNVT_Be_Time { SNVT_time_hour Bat1 SNVT_time_hour Bat2 }	**Electric heater working time** Updated every working hour. Stroed in EEPROMevery 10 working hours Reset to 0 EEPROM : nviRequest = RQ_OVERRIDE

IX.3. REGULATOR OBJECT



Config Prop	Type	Description
ncRcvHrtBt	UCPTkarnoCfg	Not Used
nciSndHrtBt	SCPTmaxSendTime	**Heartbeat delay** In seconds By default : 0
ncOffsetTemp	SCPToffsetTemp	**Offset applied on the space temp sensor (analogic or room sensor)** Range: -10 to 10 °C By default = 0
nciSetpoints	SCPTsetPnts { Signed long **occupied_cool** Signed long **standby_cool** Signed long **unoccupied_cool** Signed long **occupied_heat** Signed long **standby_heat** Signed long **unoccupied_heat** }	**Space temp setpoints depending on occupancy modes.** occupied_cool : By default = 23 °C standby_cool : By default = 25 °C unoccupied_cool : By default = 28 °C occupied_heat : By default = 21 °C standby_heat : By default = 19 °C unoccupied_heat : By default = 16 °C min = 10 °C - max = 35 °C.

ncRegulOcc	UCPT_Regul_Occ_DL { **TcndOcc :4** **GtbOcc :4** Unsigned short **TcndUnocc** Unsigned short **Reserved** Struct OccupationGtb[3] { Struct PieceOcc { **CmdLumG1** :1 **CmdLumG2** :1 SNVT_time_sec **ToPresence** SNVT_occupancy **OccVal** Unsigned short **Reserved6** } Struct PieceUnocc { **CmdLumG1** :1 **CmdLumG2** :1 SNVT_occupancy **OccVal** Unsigned short **Reserved6** } } }	**TcndOcc:** Behaviour of the room when pressing occupancy button: **Bit 0= 0:** occupancy mode is reinitialised. **Bit 0= 1:** occupancy mode is reinitialised during **ToPresence**. *Caution: when nviOccGtb=Unocc, the device works as bit0=1.* By default : 0 **GtbOcc :** Behaviour of the room occupancy when BMS occupancy changes. **0 :** OccGtb not related with OccRoom **1 :** If OccGtb goes Occ, OccRoom goes Occ (=TCND Occ) **2 :** If OccGtb goes Unocc, OccRoom goes Unocc (=TCND Unocc) **3 :** 1 and 2 combined By default : 2 **TcndUnocc** Presence detection inhibition timer (in sec). (0 = infini). By default : 1sec **OccupationGtb[3]:** Configuration of the light and sunblind behaviour during the various occupancy modes OccupationGtb[2]: used when BMS occupancy is OC_STANDBY OccupationGtb[1]: used when BMS occupancy is OC_UNOCC OccupationGtb[0]: used when BMS occupancy is OC_OCC (or any other value) **PieceOcc:** When room is occupied: - If **CmdLumG1** = 0 or 10 → All Lights of « window side » group are switched off (0) or on (10). By default : 10 - If **CmdLumG2** =10 → like CmdLumG1 for "corridor side" group **TOPresence:** Time during which the room is considered as occupied after a presence detection By default : 900 sec **OccVal:** Value of the variable nvoEffectOccup during this occupancy state **Reserved6** : Not used **PieceUnocc:** Same as PieceOcc but when the room is unoccupied (no ToPresence running)

ncCfgFan	UCPTcfgFan { Unsigned short **mode** Unsigned short **cfg** Unsigned short **override** Unsigned short **level1** Unsigned short **level2** Unsigned short **level3** Unsigned short **mini** Unsigned short **manuf1** }	**Fan control parameters:** **.mode** : 0 - No 0-10 V fan 1 - 3 fan speeds levels on 0-10 V 2 - variable fan speed on 0-10 V 3 - fan speed based on occupancy **.cfg** 0 normal 1 no fan 2 fan in cool mode only 3 fan in heat mode only **.override** 0 no forcing 1 mini V1 if occupancy is not unoccupied 2 mini V1 if occupancy is not unoccupied but stop command is allowed 3 mini V1 for every occupancy mode 4 same as 2 with V1 during 5 minutes every 2 hours if unoccupied **.level1** : regulation output level corresponding to V1 **.level2** : regulation output level corresponding to V2 **.level3** : regulation output level corresponding to V3 **.mini** : minimum variable fan output level **.manuf1 :** not used By default ={2,0,0,5,33,66,0,0}

ncCfgSrc	UCPTcfgScc { Unsigned short **fcctype** Unsigned short **roomModuleType** Unsigned short **roomModuleConfig** Unsigned short **roomModuleDisplay** Unsigned short **irNumber** Unsigned short **fanOffDelay** Unsigned short **sensorSelect** Unsigned short **offsetStep** Unsigned short **extensionCfg** Unsigned short **auxCmdType** Unsigned short **manuf2** Unsigned short **manuf3** }	.**fcctype** (3) See detailled types chapter IV.1.9 By default : fcctype = 3 : 2 pipes + electric heater .**roomModuleType** (0) (Not used) 0 boitier RJ-9 .**roomModuleConfig** (0) 1 room sensor unoccupancy locked. .**roomModuleDisplay** (0) LCD room sensor display 0 Setpoint offset 1 Room sensor space temp 2 Effect setpoint .**irNumber** (0) Remote controller number If irNumber =0 then RIR object accepts any remote. If irNumber =n (different from 0) then RIR object accepts only remotes configured for zone n. Range : 0..30. By default = 0. .**fanOffDelay** (180) Post-fan duration (in seconds) By default 180 s (Range 10 to 255 s) **sensorSelect** (0) Space temp origin 0 Analogic sensor 1 RJ-9 input 2 Network sensor .**offsetStep** (50) Setpoint offset step .**extensionCfg** (0) Not Used .**auxCmdType** (0) nviAuxCmd information type 1 = Presence 2 = Dew point 3 = Change Over 4 = Flow control 5 = Direct 0-10 V control **manuf2** : not used **manuf3** : not used By default = {3,0,0,0,0,180,0,50,0,0,0,0}

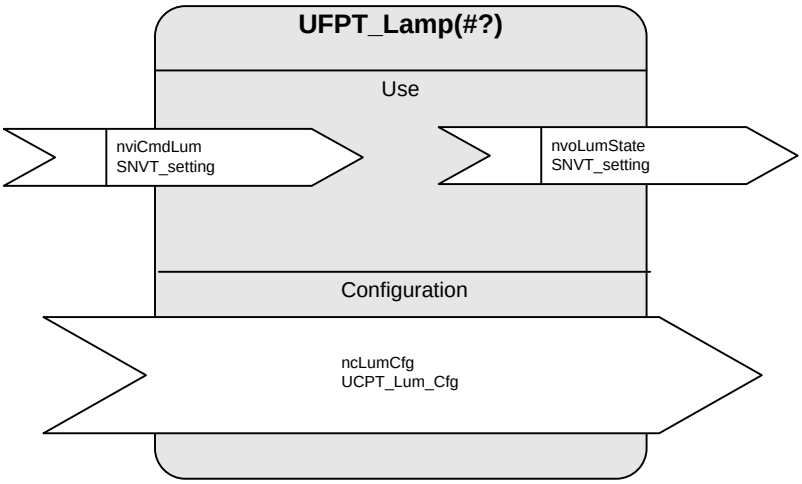
ncRelayCfg	UCPTrelayCfg { Unsigned short **type** Unsigned short **number** Unsigned short **manuf3** Unsigned short **manuf4** }	**Lx relays configuration** **.type** : 0 = network control 1 = associated to light « .number » 2 = power supply for variable speed fan **.number** : if .type = 1 only Associated light number (1 à 4) **.manuf3** : Reserved **.manuf4** : Reserved By default : {0, 0}
Variable Entrée		
nviApplicMode	SNVT_hvac_mode	**Operating mode of the controller**. HVAC_AUTO : operating mode determined by the controller HVAC_COOL : Cool mode only HVAC_HEAT or not in list : Heat mode only HVAC_OFF : Regulator stopped. Anti freeze still active HVAC_EMERG_HEAT : Anti freeze HVAC_TEST : Test mode (not used) By default = HVAC_AUTO
nviAuxCmd	SNVT_switch	**Information from master** Information type defined in ncCfgFcc.auxCmdType If ncCfgFcc.auxCmdType = 5 (0-10 V network control) **.state** = 1 **.value** : 0 = 0V, 100 = 10 V
nviCmdRelay	SNVT_switch	**Network relay command** **.state** = 1 → relay ON, = other value → relay OFF
nviEnergyHoldOff	SNVT_switch	**Energy saving mode** For window contact loop. **.state** = 1 → regulator stopped **.state** = autre valeur → normal mode
nviFanSpeedCmd	SNVT_switch	**Fan speed command** 5 states :stop, V1, V2, V3, AUTO. **state** / **value** / **Description** 0 / n/a / Stop 1 / 0% / Stop 1 / 33% / V1 1 / 66% / V2 1 / 100% / V3 0xFF / n/a / AUTO : Speed defined by the regulation loop By default : nviFanSpeedCmd = {0,0xFF} = AUTO

nviLoopWind	SNVT_switch	**Energy saving chaining variable** Looping for window contact management (master/slave configuration only) **.state** = 1 → Regulator stopped **.state** = autre valeur → normal mode By default : nviLoopWind = {0,0xFF}
nviOccManCmd	SNVT_occupancy	**BMS occupancy command.** **Cancels every forcing when updated** Range : OC_OCCUPIED, OC_UNOCCUPIED, OC_NUL, OC_STANDBY Invalid value = 0xFF = OC_NUL processed as OC_OCCUPIED By default = OC_OCCUPIED
nviOverrideOcc	SNVT_occupancy	**Room occupancy forcing information** (master/slave configuration) Occupancy information from master's nvoPresence. Range : OC_OCCUPIED, OC_UNOCCUPIED, OC_NUL Invalid value = 0xFF = OC_NUL processed as OC_UNOCCUPIED By default = OC_NUL
nviSetpoint	SNVT_temp_p	**Central setpoint information** Sets the central setpoint (middle of dead zone). The regulator updates the heat and cool setpoints values. min : 10 °C, max 35 °C By default : 327,67 (invalid)

nviSetptOffset	SNVT_temp_p	**Setpoint offset information** en °C. Not evaluated for OC_UNOCCUPIED. Range : -10 °C to 10 °C Invalid value = 0x7FFF = 327,67 °C, processed as 0 °C By default = 0 °C
nviSpaceTemp	SNVT_temp_p	**Network space temp** Used in priority over other sensors if valid. Range : -10 °C to 65 °C By default = 327,67 (invalid)
Variable Sortie		
nvoAuxContact	SNVT_switch	**Auxilliary contact state** By default : {0,0} for opened state {100,1} pour closed state
nvoCmdLum	SNVT_setting	**Light command variable** With factory settings (no bindings), it controls lights 1 et 2 of the B2L connected on the RJ-9 input of the regulator functional block.
nvoEffectOccup	SNVT_occupancy	**Occupancy state used for HVAC regulation** Calculated from nviOccManCmd and nvoPresence. Configurable value by setting cpRegulOcc for each occupancy mode.
nvoEffectSetpt	SNVT_temp_p	**Effective setpoint** used by HVAC regulation. Calculated by the regulator
nvoEnergyHoldOff	SNVT_switch	**Energy saving information** Synthesis of energy saving commands. Used for master slave configurations. **.state** = 1 → regulator stopped **.state** = autre valeur → normal mode
nvoFanSpeedCmd	SNVT_switch	**Fan speed command** state value Description ========== 0 0 % Arrêt 1 33 % V1 1 66 % V2 1 100 % V3 -1 n/a Automatic
nvoHeatCool	SNVT_hvac_mode	**Effective regulation mode.** Range : HVAC_HEAT, HVAC_COOL, HVAC_OFF, HVAC_EMERG_HEAT, HVAC_AUTO, HVAC_MRNG_WRMUP By default: HVAC_OFF
nvoMainContact	SNVT_switch	**Main contact state** By default : {0,0} for contact opened {100,1} for contact closed

nvoOccManCmd	SNVT_occupancy	**BMS occupancy state**
nvoPresence	SNVT_occupancy	**Room occupancy state** Occupancy forcing or presence detection. By default : OC_NUL
nvoSetptOffset	SNVT_temp_p	**Setpoint offset** In °C
nvoSpaceTemp	SNVT_temp_p	**Space temperature** Used by HVAC regulation In °C
nvoUnitStatus	SNVT_hvac_status { Hvat_t **mode** Signed long **Heat_output_primary** Signed long **Heat_output_secondary** Signed long **Cool_output** Signed long **Econ_output** Signed long **Fan_output** Unsigned_short **in_alarm** }	**Regulator status.** **.mode** Regulator mode **.heat_output_primary** primary heat output command in percent .**heat_output_secondary** secondary heat output command in percent **.cool_output** cool output command in percent **.econ_output** not used **.fan_output** effective fan speed **.in_alarm** defect presence (0 = no defect) By default: {HVAC_OFF,0,0,0,0,0,0}

IX.4. LIGHT OBJECT



x 4

Bloc Lampe :

1 nvi
\+ 1 nvo

2 var

Variable	Type	Description
Configuration parameter		
cpLumCfg	**UCPT_Lum_Cfg** **{** Unsigned short **LightMode** Unsigned short **Group** Unsigned short **MinGrada** Unsigned short **MaxGrada** SNVT_time_hour **MaxLightTime** Unsigned short **VitGradaLum** Unsigned short **Connection** Unsigned short **Reserved1** **}**	**LightMode: TOR** Not used – Do not modify By default : 0 **Group:** Light group (for remote sensor) Range : 0 (every groups) to 8 **MinGrada, MaxGrada:** Not used – Do not modify Defaut : 0 and 100 % **MaxLightTime:** Not used By default : 4000 heures **VitGradaLum :** Not used By default : 20 %/sec **Connection:** Identifier of lamp to command (see V.2.2) **Reserved1:** reserved
cpLumRegul	**UCPT_Lum_Regul** **{** Unsigned short Regul_OnOff Unsigned short Regul_Group Unsigned short MinRegul Unsigned short Reserved1 **}**	**Regul_OnOff:** Not used – Do not modify By default : 1 **Regul_Group:** Regulation groupe : process commands for : « Window side » : 1 « Corridor side » : 2 Both sides : 0 **MinRegul:** Not used By default : 0 **Reserved1:** reserved
Entrée		
nviCmdLum	**SNVT_setting**	**Light command input variable.**
Sortie		
nvoFbLum	**SNVT_setting**	**Master slave chaining variable** Actual command in 'rotation' field Actual light state in 'setting' field.

IX.4.1. Light commands

nviCmdLum	Source	Description
SET_STATE, 100, –(10+Group)	user	Forcing ON (on/off lamp) or dimming UP (dimming lamp).
SET_STATE, 0, –(10+Group)	user	Forcing OFF (on/off lamp) or dimming DOWN (dimming lamp).
SET_STOP, 0, –(10+Group)	user	Forcing light, end of the dimming up or down command
SET_STATE, state, N%	Master light	For the master/slave chaining.

nvoFbLum	Description
SET_STATE, State, N%	The lamp is dimmed at N %. The lamp state is in the setting field.

Light states :

State (nvoFbLum.setting)	Description
0	Reserved
1	1 Lamp not active.
2	2 Lamp is forced by the user or the BMS
3	3 Lamp is blocked by the BMS
4	4 Lamp is dimming up
5	5 Lamp is dimming down
6	6 Lamp is steady
7	7 Lamp is controlled by the control loop
10	10 Lamp is a slave

X. DOCUMENT HISTORY

Indice document	Date	Rédacteur	Modifications	Version
0.1	12/05/09	V.MACABIES	Initial version.	1.00
1.3	30/03/11	M.Habenicht	Modifications for Saia Burgess Controls	1.30
2.0	02/09/13	M.Habenicht	New company name and company logo	

Address of Saia-Burgess Controls AG

Saia-Burgess Controls AG

Bahnhofstrasse 18
3280 Murten / Switzerland

Tel: +41 26 672 72 71
Fax: +41 26 672 74 99
E-mail: support@saia-pcd.com
Homepage: www.saia-pcd.com
Support: www.sbc-support.com

Postal address for returns from customers of the Swiss Sales office:

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