



PZM2114 Software Users Guide
Version 2.06 and 3.00
Draft 3 Feb. 2004

| | | |
|----------|---|-----------|
| 1 | Software functions..... | 4 |
| 1.1 | General purpose function profiles:..... | 5 |
| 1.1.1 | Occupancy Sensor (1060), 2 objects..... | 5 |
| 1.1.2 | Occupancy Controller (3071), 2 objects | 8 |
| 1.2 | Lighting profiles:..... | 12 |
| 1.2.1 | Light Sensor Input (1010), 1 objects..... | 12 |
| 1.2.2 | Lamp Actuator (3040), 5 objects | 15 |
| 1.2.3 | Constant Light Controller (3050), 2 objects ONLY version 3.00 | 19 |
| 1.3 | Space Comfort profiles: | 25 |
| 1.3.1 | HVAC Temperature Sensor Input (1040), 1 objects..... | 25 |
| 1.3.2 | Space Comfort Controller (8500), 1 objects | 27 |
| 1.3.3 | Heat Actuator, 1 objects..... | 36 |
| 1.3.4 | Cool Actuator, 1 objects..... | 39 |
| 1.3.5 | Ventilation Actuator CAV, 1 objects..... | 41 |
| 1.3.6 | Ventilation Actuator VAV, 1 objects..... | 43 |
| 1.4 | Sunblind profiles:..... | 45 |
| 1.4.1 | Sunblind Actuator, 2 objects..... | 45 |
| 2 | Applications and bindings | 47 |
| 2.1 | Manual Control, Light Bindings | 47 |
| 2.1.1 | Simple on and off..... | 47 |
| 2.1.2 | Simple on and off with feed back. | 47 |
| 2.1.3 | Multi on and off with feed back..... | 47 |
| 2.1.4 | Simple on and off with multi actuators 1..... | 47 |
| 2.1.5 | Simple on and off with multi actuators 2..... | 47 |
| 2.1.6 | Multi on and off with multi actuators 1. | 48 |
| 2.1.7 | Multi on and off with multi actuators 2. | 48 |
| 2.2 | Occupancy automatic, Light Bindings..... | 48 |
| 2.2.1 | Occupancy on off..... | 48 |
| 2.2.2 | Occupancy on off, with manual override..... | 48 |
| 2.2.3 | Occupancy on off, manual override and manual control LED. | 49 |
| 2.2.4 | Occupancy and neighbour occupancy on off..... | 49 |
| 2.2.5 | Occupancy, neighbour occupancy on off and manual override..... | 49 |
| 2.3 | Constant light automatic, Light Bindings | 50 |
| 2.3.1 | Constant light basis automatic | 50 |
| 2.3.2 | Constant light with slave output..... | 50 |
| 2.3.3 | Constant light and occupancy control..... | 50 |
| 2.3.4 | Constant light, occupancy and manual control | 50 |
| 2.3.5 | Constant light, dual occupancy and a manual control | 51 |
| 2.4 | Space Comfort..... | 52 |
| 2.4.1 | Heating | 52 |
| 2.4.2 | Cooling..... | 52 |
| 2.4.3 | Ventilation with occupancy control | 52 |
| 2.4.4 | Heat and cool with occupancy automatic | 53 |
| 2.4.5 | Heat, cool and ventilation with occupancy automatic | 53 |



3 Figur list..... 54



1 Software functions.

The following section is describing the software function profiles implemented in the Lonbox Comfort and Light Controller model PZM2114.

The PZM2114 is designed to fulfil the requirement for operating and controlling lighting, heating, cooling, ventilation and sunblind or windows in modern Intelligent Building Installations, using the LonWorks network standard ANSI/EIA/CEN 709.1 or as a stand-alone unit.

1.1 General purpose function profiles:

The following is describing the general purpose software functions implemented in the Lonbox Comfort and Light Controller model PZM2114.

1.1.1 Occupancy Sensor (1060), 2 objects

The following describes the function for the occupancy sensor object. This object is used with a hardware sensor whose output is either in an occupied or unoccupied state. The occupancy sensor object is used to detect occupancy in a room or an area and keep the occupied state until no occupancy can be detected. The output from the occupancy sensor object is typically connected to a controller. The occupancy controller takes care of the proper action and calculates application delay or hold times as appropriate.

| PIR # | Connector # | Name in LonMaker list |
|-------|-------------|-----------------------|
| 1 | Input 2 | OccupSensor_1 |
| 2 | Input 3 | OccupSensor_2 |

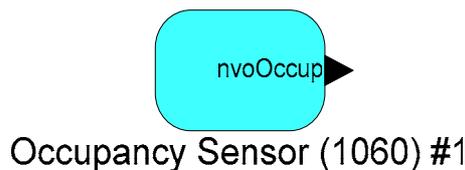


Figure 1 The occupancy sensor object

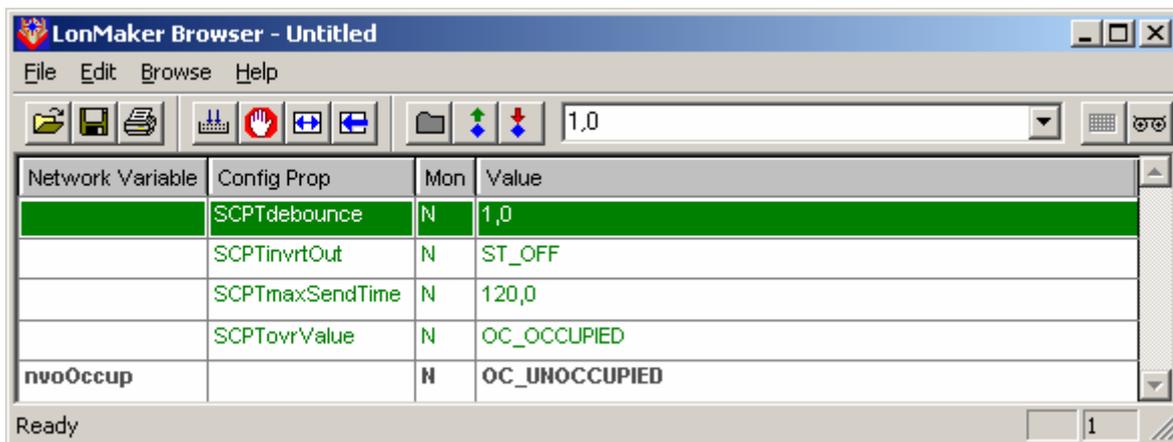


Figure 2 Browsing the occupancy sensor object

1.1.1.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|---------------|---------------|----------------------|---------------|
| nvoOccup | OC_UNOCCUPIED | SVNT_occupancy (109) | Sensor output |

1.1.1.1.1 nvoOccup

This output network variable provides the qualified state of the hardware sensor output connected to the Comfort and Ligth Controller. The sensor input can be inverted with the configuration property SCPTinvtOut.

The variable is transmitted when the occupancy state changes from unoccupied to occupied or when the occupancy state has changed from occupied to unoccupied at the frequency of the heartbeat. The maximum update rate is fixed by the heartbeat frequency and the minimum update rate is fixed by the debouncing time. The default service type is acknowledged.

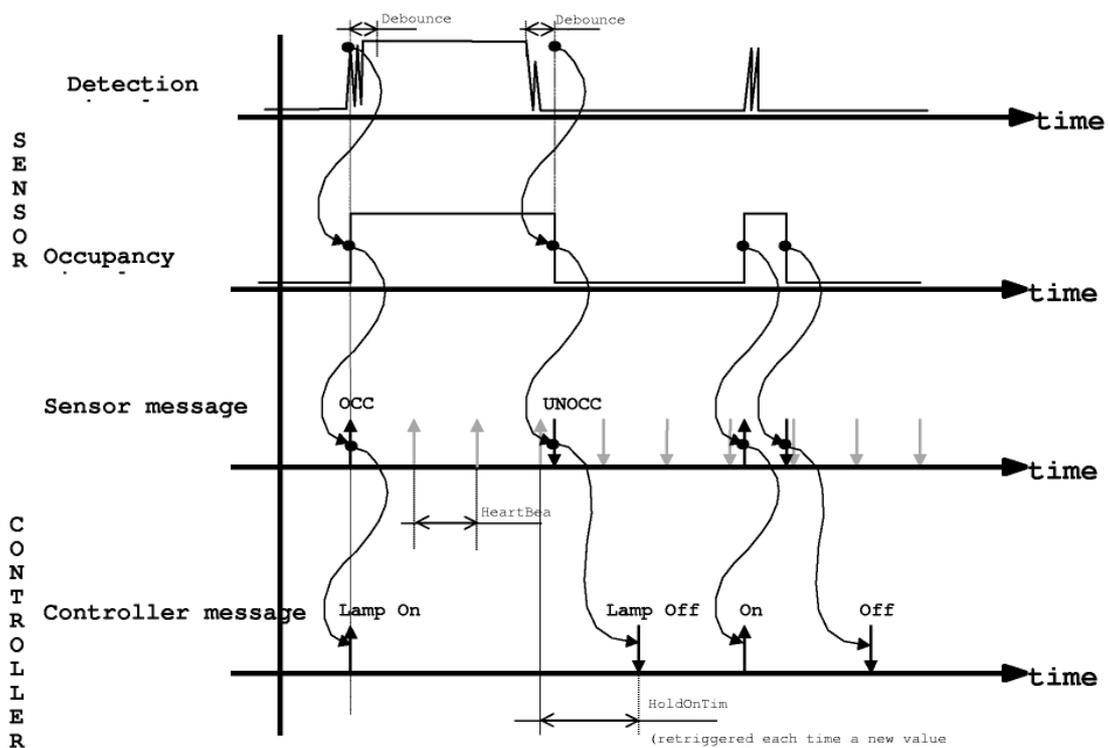
1.1.1.2 Configuration Properties

| Configuration | Default Value | Description |
|----------------------|---------------|-----------------------------|
| SCPTdebounce (139) | 0,0 | Debounce filter time |
| SCPTinvtOut (16) | ST_OFF | Invert the output |
| SCPTmaxSendTime (49) | 120,0 | Heartbeat time |
| SCPTovrValue (33) | OC UNOCCUPIED | State when in override mode |

1.1.1.2.1 SCPTdebounce

This mandatory configuration network variable defines the debounce time to generate the detection envelop and the OCCUPIED and NON OCCUPIED messages. Valid range is 0.0 - 6553.4 by steps of 0.1s.

The following timing diagram defines the different timers used in the profile, it illustrate the debounce filter function.





1.1.1.2.2 SCPTinvertOut

Invert the input signal. This function can be used with NC occupancy sensors.

1.1.1.2.3 SCPTmaxSendTime

SCPTmaxSendTime is a configuration property. This configuration defines the repeat period between to value update sent on the bus also called a heartbeat. The aims of the heartbeat is to be sure that the sensor is alive and to permit a controller to have multiple sensors on the same input

Remark : Even if the heartbeat is optional, the heartbeat functionality shall be implemented in the Occupancy Sensor with a 2 minutes repeat period, (default value).

1.1.1.2.4 SCPTovrValue

This override value sets the value the actuator should adopt when the object is overridden and behavior is OV_SPECIFIED.

1.1.2 Occupancy Controller (3071), 2 objects

The following section is describing the occupancy controller. Typically the occupancy controller input is connected to the occupancy sensor and the output to a lamp. A switch can be used to turn the occupancy controller into ON and OFF mode. An additional switch can override the controller and directly control the lamp. The occupancy controller can also be driven by an occupancy sensor from a neighboring area, secondary input. The secondary function gives the users a secure feeling, due to the light around. The occupancy controller can also operate another controller, eg. a constant light controller.

| Controller# | Name in LonMaker list |
|-------------|-----------------------|
| 1 | OccupancyCtrl[0] |
| 2 | OccupancyCtrl[1] |

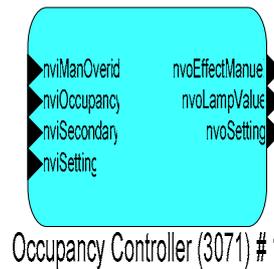


Figure 3 The Occupancy controller object

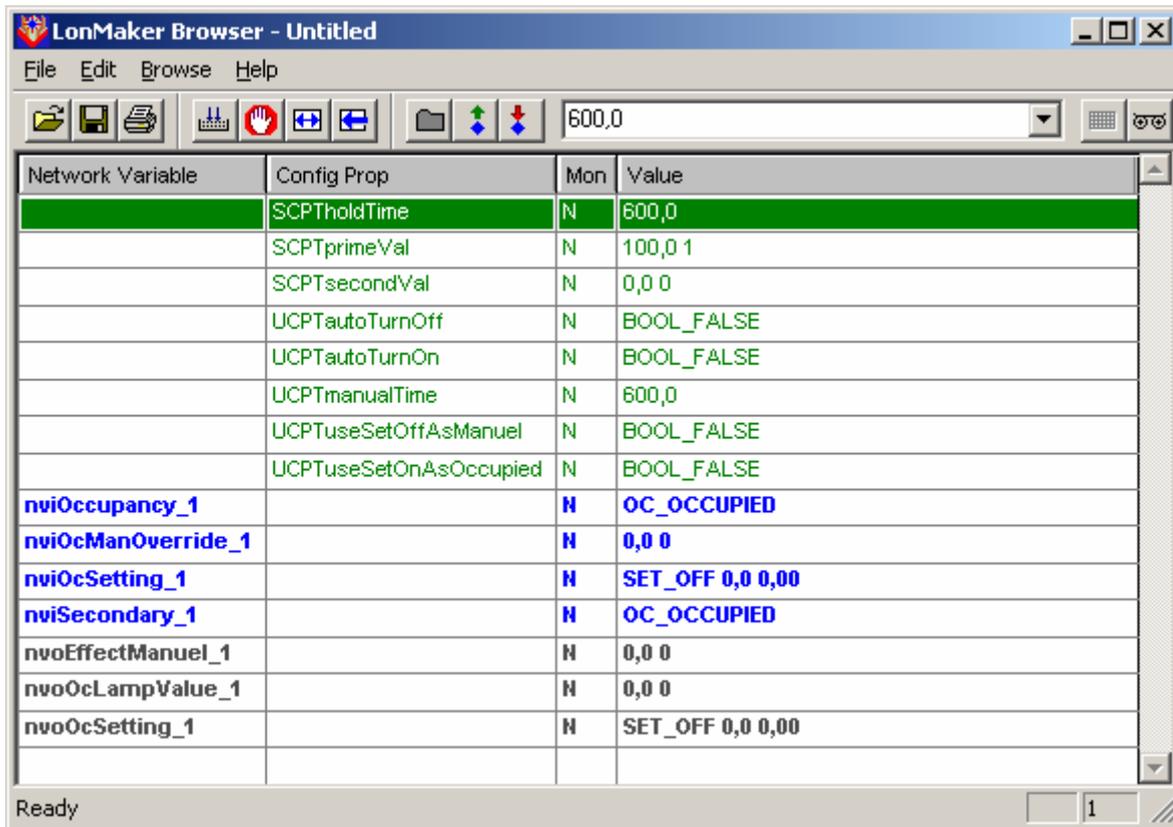


Figure 4 Browsing the occupancy controller object

1.1.2.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|------------------|------------------|----------------------|-------------------------------|
| nviOcManOverride | 0,0 0 | SNVT_switch (95) | Manual Operation |
| nviOccupancy | OC_OCCUPIED | SVNT_occupancy (109) | Occupancy input |
| nviSecondary | OC_OCCUPIED | SVNT_occupancy (109) | Neighbourhood Occupancy input |
| nviOcSetting | SET_OFF 0,0 0,00 | SNVT_setting (117) | Setting control input |
| nvoEffectManuel | 0,0 0 | SNVT_switch (95) | Manual indicator output |
| nvoOcLampValue | 0,0 0 | SNVT_switch (95) | Controller output |
| nvoOcSetting | SET_OFF 0,0 0,00 | SNVT_setting (117) | Setting control output |

1.1.2.1.1 nviManOverride

Manual override input, enables the local and manual control of the lamp value output.

1.1.2.1.2 nviOccupancy

Occupancy status input value, provides the occupancy status for the area.

1.1.2.1.3 *nviSecondary*

Secondary occupancy status input value. It provides the occupancy status of a neighbourhood area, in order to provide low-level lighting around an occupied area for a feeling of security.

1.1.2.1.4 *nviSetting*

Setting input (Auto/Off). This mode can be either be ON(AUTO) or OFF.

1.1.2.1.5 *nvoEffectManuel*

This output variable reflect the status of manual override.

1.1.2.1.6 *nvoLampValue*

Lamp value output. State for the lamp actuator ON or OFF, and the percentage level of intensity.

1.1.2.1.7 *nvoSetting*

Setting output. Select the operating mode for another controller, e.g., constant light controllers.

1.1.2.2 Configuration Properties

| Configuration | Default Value | Description |
|------------------------|---------------|-----------------------------------|
| SCPTholdTime (91) | 600 | Occupancy delay timer |
| SCPTprimeVal (155) | 100,0 1 | Output value when occupied |
| SCPTsecondVal (156) | 60,0 1 | Output for Value neighboring |
| | | |
| UCPTautoTurnOff | BOOL TRUE | Turn off selection, true or false |
| UCPTautoTurnOn | BOOL TRUE | Turn on selection, true or false |
| UCPTmanualTime | 600 | Delay timer when manual control |
| UCPTuseSetOffAsManuel | BOOL TRUE | Use OFF as manual |
| UCPTuseSetOnAsOccupied | BOOL TRUE | Use ON as occupied |

1.1.2.2.1 *SCPTholdTime*

Hold time for occupied state after there is no occupancy detected. This timer is used for both the main occupancy input and for the secondary input.

1.1.2.2.2 *SCPTprimeVal*

The default output value when an area is occupied.

1.1.2.2.3 *SCPTsecondVal*

The default output value when the neighboring area is occupied, when the secondary occupancy sensor input variable is active.

1.1.2.2.4 UCPTautoTurnOff

This configuration property is used to select if the controller automatically switch off the output when it time out the timer.

| UCPTautoTurnOff | Description |
|-----------------|----------------------------|
| BOOL_TRUE | Automatic off activated |
| BOOL_FALSE | No automatic off activated |

1.1.2.2.5 UCPTautoTurnOn

This configuration property is used to select if the controller automatically switch on the output when occupancy is detected.

| UCPTautoTurnOn | Description |
|----------------|---------------------------|
| BOOL_TRUE | Automatic on activated |
| BOOL_FALSE | No automatic on activated |

1.1.2.2.6 UCPTmanualTime

This configuration property is used to set the timer value used during a manual override input on the nviManOverride. The manual timer is also serviced by the occupancy input and will be set every time an occupancy signal is received. Hereafter the timer will countdown and when it times out manual override will be deactivated.

1.1.2.2.7 UCPTuseSetOffAsManuel

Use SET_OFF as manual.

| UCPTuseSetOffAsManuel | Description |
|-----------------------|-----------------------------|
| BOOL_TRUE | Use SET_OFF as manual |
| BOOL_FALSE | Don't use SET_OFF as manual |

1.1.2.2.8 UCPTuseSetOnAsOccupied

Use SET_ON as occupied.

| UCPTuseSetOffAsManuel | Description |
|-----------------------|------------------------------|
| BOOL_TRUE | Use SET_ON as Occupied |
| BOOL_FALSE | Don't use SET_ON as Occupied |

1.2 Lighting profiles:

The following is describing the lightning software functions implemented in the Comfort and light controller Lonbox PZM2114.

1.2.1 Light Sensor Input (1010), 1 objects

The following describes the light sensor object. The object is used to measure ambient light levels. The light sensor object is used together with controller objects, such as the constant light controller. Typically the light sensor output is connected to the constant light controller input.

The light object service the hardware input used for light sensors and convert the hardware signals to a digital SNVT resource on the network. The hardware connected is detected as a 0 til 10 voltage input signal.

| LUX # | Connector # | Name in LonMaker list |
|-------|-------------|-----------------------|
| 1 | Input 1 | LightSensor |



Light Sensor (1010) #1

Figure 5 The light sensor object

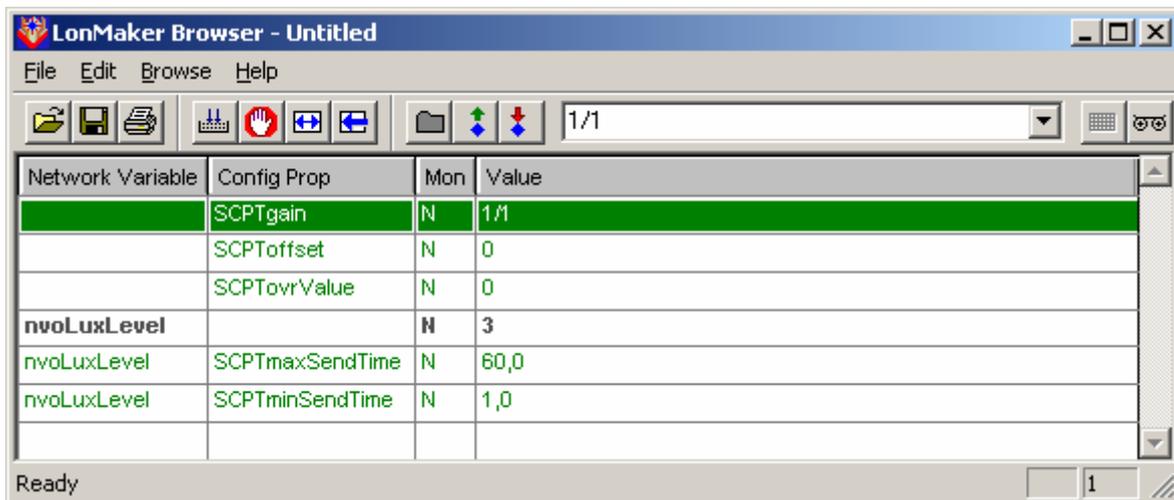


Figure 6 Browsing the light sensor object

1.2.1.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|---------------|---------------|---------------|--------------------|
| nvoLuxLevel | 0 | SNVT_lux (79) | Output light level |

1.2.1.1.1 *nvoLuxLevel*

This variable is the output from the light sensor object. The hardware monitors the 0 to 10 volt input voltage and send out the variable manipulated by the configuration properties UCPTsensorOffset and UCPTsensorRange

1.2.1.2 Configuration Properties

| Configuration | Default Value | Description |
|----------------------|---------------|----------------------------|
| SCPTovrValue (33) | 0 | Lux level when overwrite |
| SCPTmaxSendTime (49) | 60,0 | Maximum receive time |
| SCPTminSendTime (52) | 1,0 | Minimum send time |
| | | |
| UCPTgain | 1/1 | Lux level output range |
| UCPTOffset | 0 | Lux level offset from zero |

1.2.1.2.1 *SCPTovrValue*

This configuration property set the value for the output variable, when the object is in override mode. The variable type is a SNVT_lux (79). The value is used on the nvoLuxLevel variable from the object.

1.2.1.2.2 *SCPTmaxSendTime*

This configuration property set the value for the send heartbeat. Maximum period of time that expires before the object automatically transmits the present value of the lux level output NV.

1.2.1.2.3 *SCPTminSendTime*

This configuration property set the value for the minimum send time. Minimum period between output NV transmissions (maximum transmission rate).

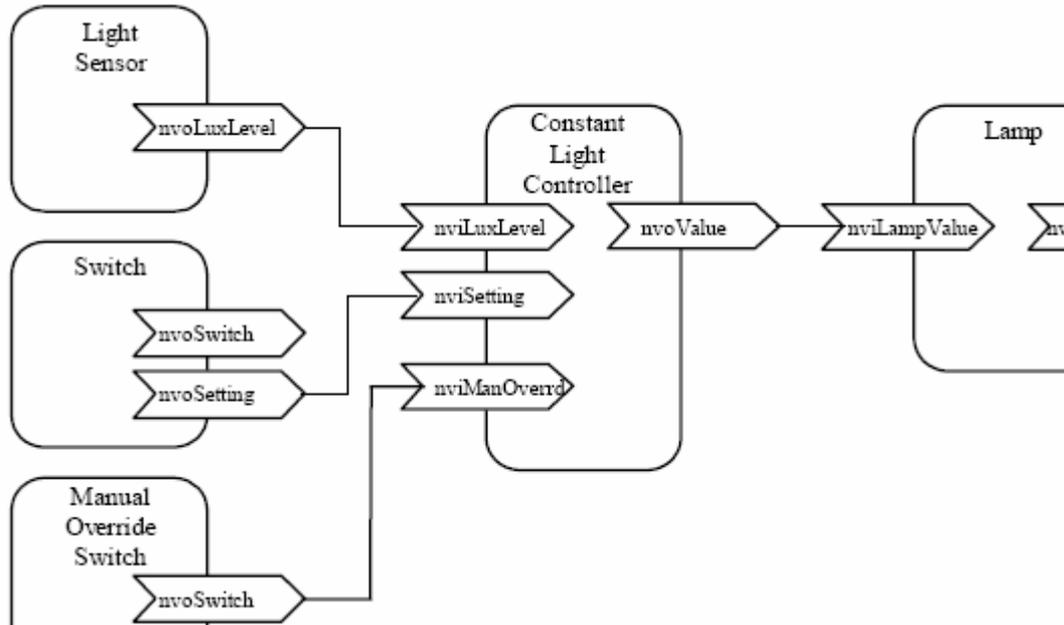
1.2.1.2.4 *UCPTgain*

This configuration property is used to calibrate the external hardware input. It will calibrate the sensor range output. The input voltage will be gain with the value.

1.2.1.2.5 *UCPToffset*

This is used to calibrate the external hardware input. It will calibrate the sensor ra is used to calibrate the external hardware input. It will calibrate the sensor offset failure from the value 0. If the value is set to 20, 0 volt on the input will be represented as 20 lux on the network and the lowest value send out is 20 lux.

1.2.1.3 Light sensor object typically use



1.2.2 Lamp Actuator (3040), 5 objects

The following describes the lamp actuator object. The lamp actuator object is used to control the illumination level of a hardware lamp output. The lamp actuator is used with switch and controller devices such as the constant light controller and scene controller. A switch object output nvoSwitch is connected to the input nviLampValue of the lamp actuator. Controller objects can be used between switch type sensors and lamp actuators. In cases of multiple sensors the feedback connection can be used to synchronize a group of switches.

| Lamp # | Relay connector # | 0-10 connector # | Name in LonMaker list |
|--------|-------------------|------------------|-----------------------|
| 1 | LIGHT 5 | DIMMER 1 | LampActuator[0] |
| 2 | LIGHT 4 | DIMMER 2 | LampActuator[1] |
| 3 | LIGHT 3 | Not used | LampActuator[2] |
| 4 | LIGHT 2 | Not used | LampActuator[3] |
| 5 | LIGHT 1 | Not used | LampActuator[4] |

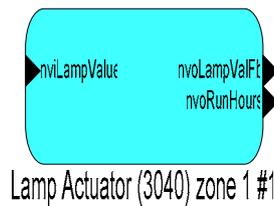


Figure 7 The lamp actuator object

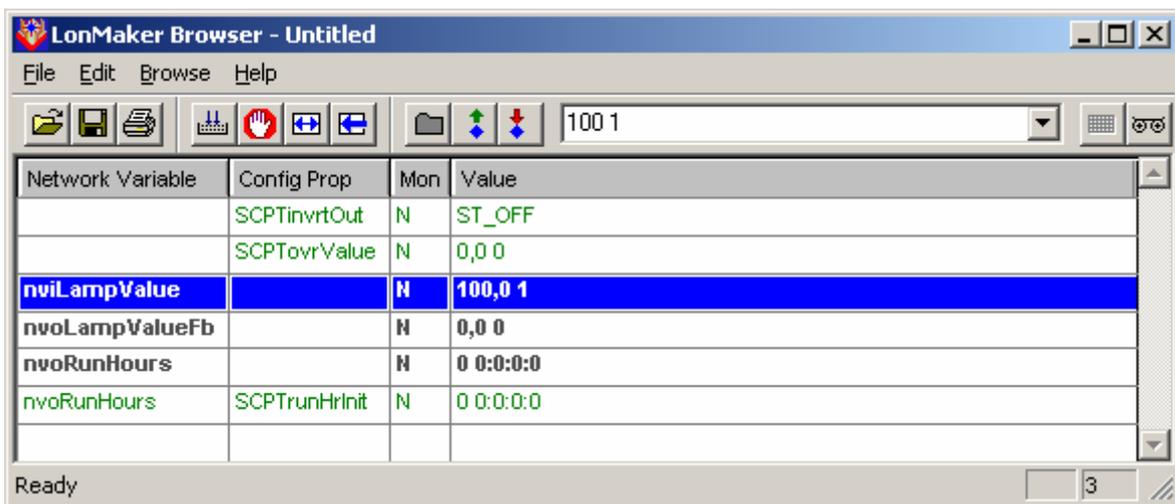


Figure 8 Browsing the lamp actuator object



1.2.2.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|----------------|---------------|----------------------|-------------------------|
| nviLampValue | 0,0 0 | SNVT_switch (95) | Control input |
| nvoLampValueFb | 0,0 0 | SNVT_switch (95) | Lamp actuator feed back |
| nvoRunHours | 0 0:0:0:0 | SNVT_elapsed_tm (87) | Elapsed ON time |

1.2.2.1.1 nviLampValue

This variable is the control input. Permits other devices to control the Lamp Actuator level and state.

1.2.2.1.2 nvoLampValueFb

Lamp feedback output variable. State of the Lamp Actuator (ON or OFF) and the percentage level of intensity.

1.2.2.1.3 nvoRunHours

This output variable contains the value for the accumulated actuator ON time. The Comfort and Light Controller automatically monitor the lamp actuator's ON time. The value can be set and reset with the configuration property SCPTrunHrInit. The timer is enabled counting time, whenever the lamp actuator is turned ON. Elapsed time have following format:

day hour : minute : second : millisecond.

It is only the day and hour field in the structure that is used. Internally the counter is counting in seconds, but it only updates the hour and date field.

1.2.2.2 Configuration Properties

| Configuration | Default Value | Description |
|---------------------|---------------|-------------------------------|
| SCPTovrValue (33) | 75,0 1 | Lamp value when overwrite |
| SCPTinvrtOut (16) | ST_OFF | Setting if output is inverted |
| SCPTrunHrInit (135) | 0 0:0:0:0 | Set or reset Running hours |

1.2.2.2.1 SCPTovrValue

This configuration type set the value for the output variable, when the object is in override mode. The variable type is a SNVT_switch (95). The value is used on the nvoLampValue variable from the object.

1.2.2.2.2 *SCPTinvrtOut*

This configuration property set the parameter, if the lamp output is inverted or not. This possibility is used when an external relay is used and the electrical installation shall be fail-safe.

| SCPTinvrtOut | Description |
|---------------------|---------------------|
| ST_OFF | Output not inverted |
| ST_ON | Output inverted |

1.2.2.2.3 *SCPTrunHrInit*

This configuration is used to initialize the running hour counter value. The counter is send out on the SNVT variable nvoRunHours.

Elapsed time have the following format: day hour : minute : second :millisecond.

It is only the day and hour field in the structure that is used. Internally the counter is counting in seconds, but it only updates the hour and date field in the structure.

To change the value it's important to change value from what it is and not only resend the value existing in the property all ready.

For example:

The configuration property value is already 0 0:0:0:0 and we want to set the nvoRunHours to 0 0:0:0:0 using the configuration property. To do this you need to change the configuration property different from 0 0:0:0:0, it could be 0 1:0:0:0. Here after we set the value back to 0 0:0:0:0 and now the nvoRunHours is reset to 0 0:0:0:0.

Another example:

We want to set the nvoRunHours to 18 days and 1 hour (18 1:0:0:0). The configuration property value is 0 0:0:0:0, we can directly enter 18 1:0:0:0 to change the nvoRunHours.

1.2.2.2.4 Lamp object typically use

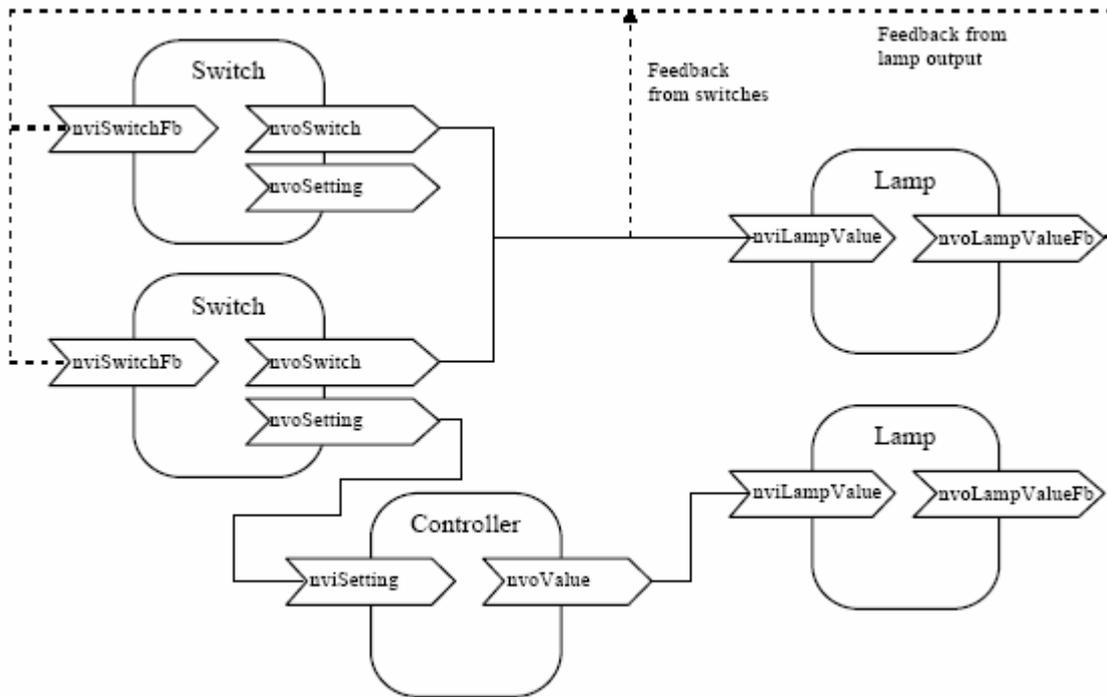


Figure 9 Typically use of the lamp actuator object

1.2.3 Constant Light Controller (3050), 2 objects ONLY version 3.00

The following section is describing constant light controller object. The controller input is the ambient light level and the output is the state and illumination level to the lamp actuator.

Typically the constant light controller input is connected to a light sensor object and the output to a lamp actuator object. An extern input can be used to turn the constant light controller object into AUTO and OFF mode. Also the illumination level set point can temporarily be adjusted upwards and downwards. An additional switch can override the controller. When manual override input is written to, the constant light controller object is turned into MANUAL mode and the data is directly passed to the lamp.

The controller is designed from the experience Prolon have got, from the many building automation installation we have delivered components for. The controller is calculating the failure between the set point and the measured light level. If the failure is more than a fixed 8% failure dead band, the lamp output will be either be increased or decreased. The correction on the lamp output will change with the speed configured by the value in the configuration parameter step value.

| Controller # | Name in LonMaker list |
|--------------|-----------------------|
| 1 | ConstantLight_1 |
| 2 | ConstantLight_2 |

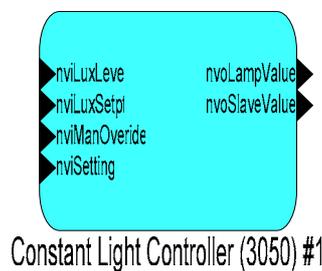


Figure 10 The constant light controller object

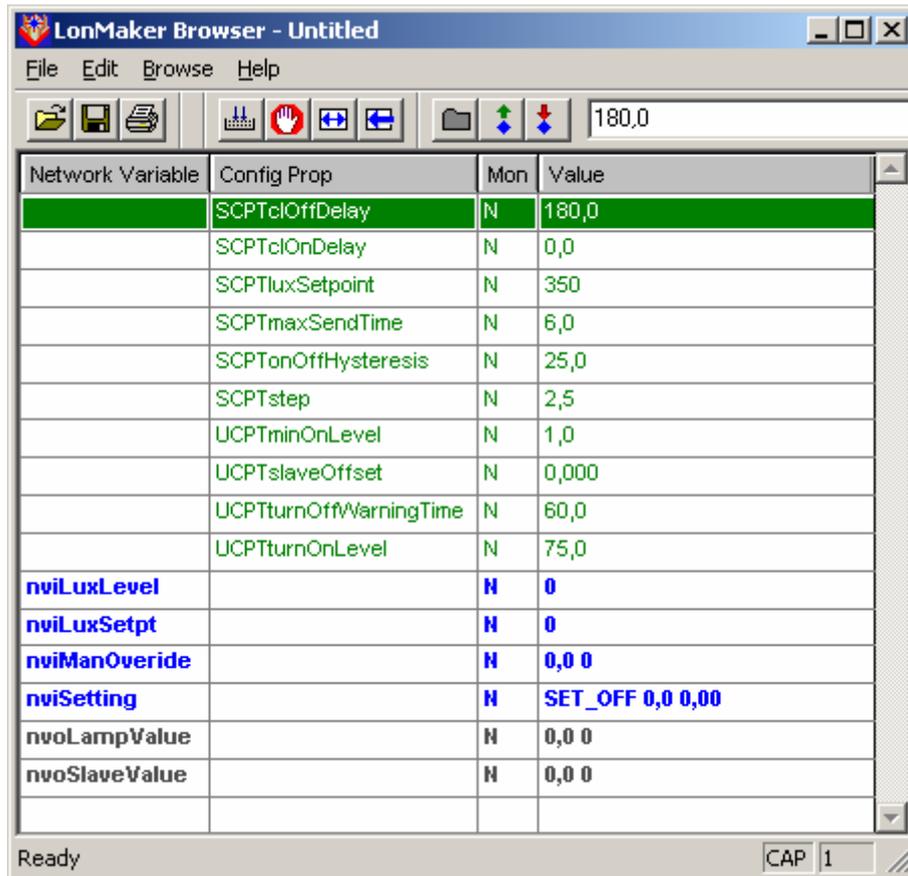


Figure 11 Browsing the constant light controller object

1.2.3.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|----------------|------------------|--------------------|----------------------------|
| nviLuxLevel | 0 | SNVT_lux (79) | Ambient light level input |
| nviSetting | SET_OFF 0,0 0,00 | SNVT_setting (117) | Mode selection, setpoint |
| nviManOverride | | SNVT_switch (95) | Manual override control |
| nviLuxSetpt | 0 | SNVT_lux (79) | Temporary set point input |
| | | | |
| nvoLampValue | 0,0 0 | SNVT_switch (95) | Lamp actuator output |
| nvoSlaveValue | 0,0 0 | SNVT_switch (95) | Slave lamp actuator output |

1.2.3.1.1 nviLuxLevel

This input variable is sourcing the ambient light level to the controller object. The constant light controller object will calculate an output based on this value and its failure from the set point defined with the SCPTluxSetpoint property or the temporary set point set by the input variable nviLuxSetpt. The primary output is the variable nvoLampValue and secondary the nvoSlaveValue.



1.2.3.1.2 nviSetting

This input network variable selects the operating mode and adjusts the setpoint of the constant light controller. Modes are ON, OFF, DOWN or UP.

The ON mode turns on the constant light controller into automatic, which then starts to control the lamp value output, so that the illumination level on the output which is provided by the nviLuxLevel equals to the setpoint value.

The OFF mode turns off the controller and the lamp value output.

The setpoint of the controller can temporarily be stepped upwards and downwards. The changes made to the setpoint value are not stored permanently into the memory. Next time when ON mode is selected the original setpoint from the SCPTluxSetpoint property is restored. The variable nviSetting uses SNVT_setting enumerations ON, OFF, UP and DOWN. With UP and DOWN functions a percentile value (0%-100% in 0.5% steps) is used to define the relative size of an increment/decrement.

1.2.3.1.3 nviManOverride

This variable is the manual override input and provides enable of manual control for the lamp value output.

1.2.3.1.4 nviLuxSetpt

This variable is the manual input for a temporary light set point. The set point will be reset to the set point defined by SCPTluxSetpoint property when the controller object is set ON with the setting input or when the Comfort and Light Controller is booted.

1.2.3.1.5 nvoLampValue

This output network variable provides the state for the lamp actuator (ON or OFF) and the percentage level of intensity. The valid range of the state is as defined for SNVT_switch where 0 means OFF and 1 means ON. The 8-bit intensity value contains a value 0 to 200, representing minimum to maximum (0%-100%) intensity. This variable is the primary output from the controller.

1.2.3.1.6 nvoSlave Value

This output network variable provides the state for the lamp actuator (ON or OFF) and the percentage level of intensity. The valid range of the state is as defined for SNVT_switch where 0 means OFF and 1 means ON. The 8-bit intensity value contains a value 0 to 200, representing minimum to maximum (0%-100%) intensity. This variable is the secondary output from the controller, normally used as a slave function. The slave output can be offset from the primary master output. Please see the following graphic to understand the offset function. The offset level can be configured with the parameter UCPTslaveOffset configuration property.

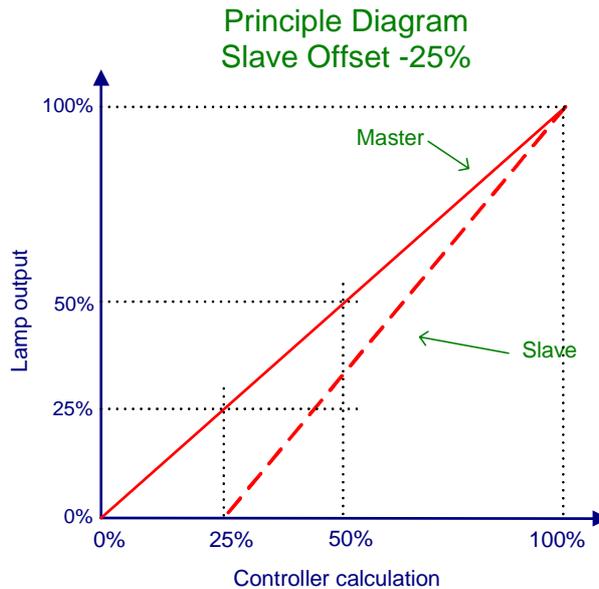


Figure 12 Understanding the light slave output

1.2.3.2 Configuration Properties

| Configuration | Default Value | Description |
|------------------------|---------------|-----------------------------------|
| SCPTclOffDelay | 180,0 | Delay time switching Off |
| SCPTclOnDelay | 0,0 | Delay time switching On |
| SCPTluxSetpoint | 350 | Controller light level set point |
| SCPTmaxSendTime (52) | 6,0 | Maximum send time |
| SCPTonOffHysteresis | 25,0 | Hysteresis between On and OFF |
| SCPTstep | 2,5 | Speed for the regulation |
| | | |
| UCPTminOnLevel | 1,0 | The lowest light level |
| UCPTslaveOffset | 0,0 | Slave offset from master |
| UCPTturnOffWarningTime | 60,0 | Warning time before switching OFF |
| UCPTturnOnLevel | 75,0 | On level for automatic regulation |

1.2.3.2.1 SCPTclOffDelay

This configuration property is used to configure the delay time, before the automatic regulation is switching OFF the output after no light is required any longer.

The light will automatic switch off, when the output from the controller is 0% light level and the required light level has been obtained compared to the light level set point. Hereafter the light will switch OFF delayed by the SCPTclOffDelay time configuration.

1.2.3.2.2 SCPTclOnDelay

This configuration property is used to configure the delay time, before the automatic regulation is starting after is has been switched off, when light is required. Light is required if the measured light



level is lower than the set point, minus the SCPTonOffHysteresis value. Hereafter the light will turn on delayed by the time, configured in the SCPTclOnDelay property.

1.2.3.2.3 SCPTluxSetpoint

This property is used to configure the light level set point defined in lux. The value can be in the range 0 to 65636.

1.2.3.2.4 SCPTmaxSendTime

This configuration property set the value for the send heartbeat. Maximum period of time that expires before the object automatically transmits the present value of the lux level output NV.

1.2.3.2.5 SCPTonOffHysteresis

1.2.3.2.6 SCPTstep

This configuration property is used to configure the speed for the regulation loop. The property describe the maximum value in % point, that the output can change in one step. The output will change every second if necessary and the maximum change in % point is defined in the UCPTstep.

1.2.3.2.7 UCPTminOnLevel

This configuration property is used to configure the minimum level for the light when the light is on. The value is described in %.

1.2.3.2.8 UCPTslaveOffset

This configuration property is used to configure the slaves offset from the master. The offset can be positive or negative related to the master lamp output. It is not a parallel offset . The value is described in %.

1.2.3.2.9 UCPTturnOffWarningTime

This configuration property is used to configure the warning time before the light turns off. This function is warning that the light soon will turn off.

1.2.3.2.10 UCPTturnOnLevel

This configuration property is used to configure the level the controller will start on when switching on in automatic mode.

1.2.3.3 Constant light controller object typically use

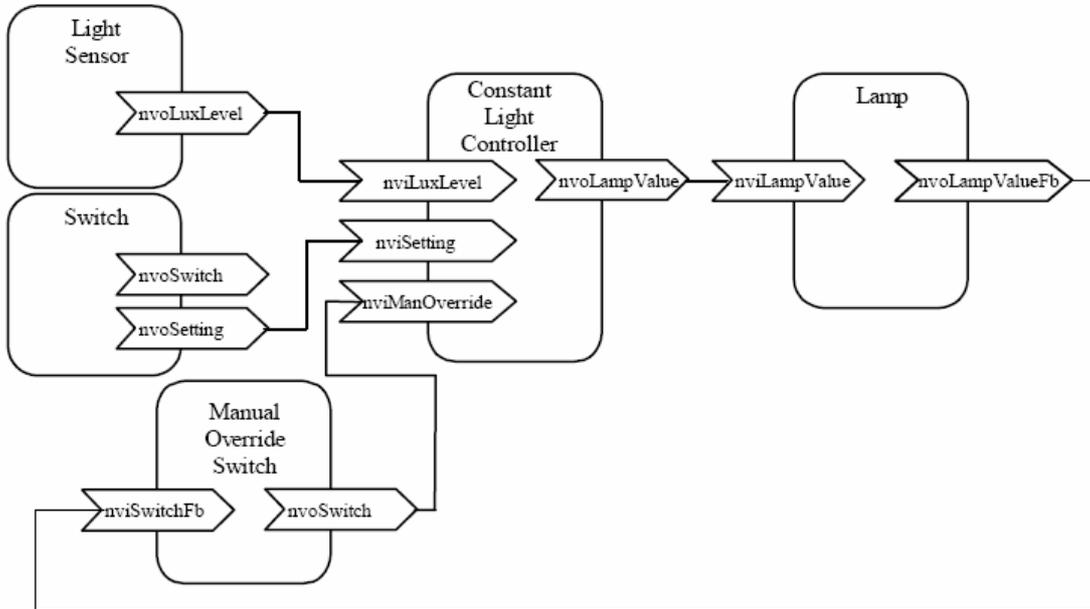


Figure 13 Constant light controller typically use

1.3 Space Comfort profiles:

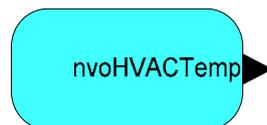
The following list is describing the space comfort software functions implemented in the Comfort and Light Controller Lonbox PZM2114.

1.3.1 HVAC Temperature Sensor Input (1040), 1 objects

The temperature sensor object is used to measure on a hardware input from a temperature sensor and make this measurement available on the network as a SNVT variable resource.

The hardware sensor connected to the input, can be one from many types as NTC I various models and PT1000. The temperature measured is often used as input to the comfort space controller object, controlling the space temperature in a zone.

| Temp # | Connector # | Name in LonMaker list |
|--------|-------------|-----------------------|
| 1 | TEMP. | HvacTempSensor |



HVAC Temperature Sensor (1040) # 1

Figure 14 The temperature sensor object

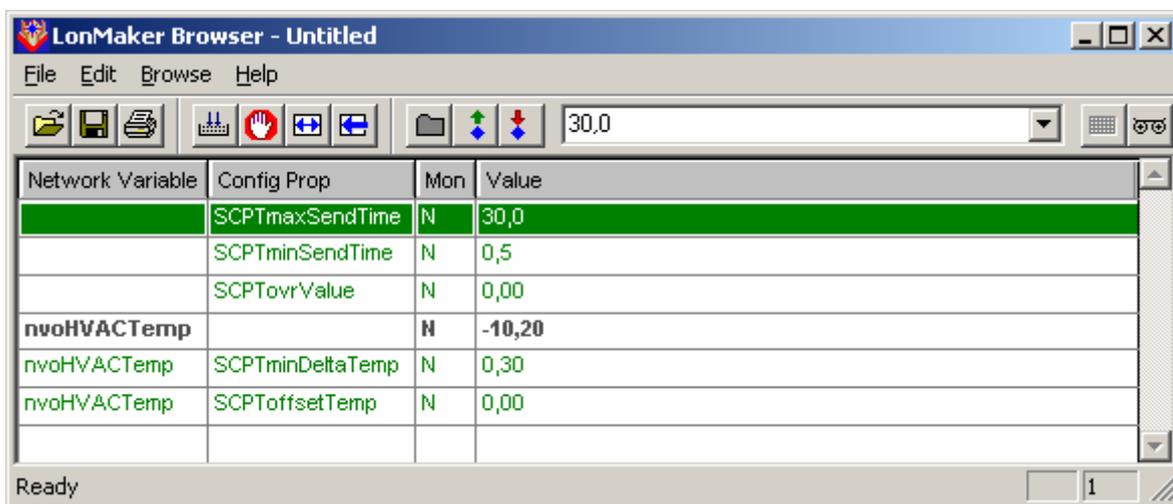


Figure 15 Browsing the temperature sensor object

1.3.1.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|---------------|---------------|------|-------------|
| nvoHVACTemp | | | |

1.3.1.1.1 nvoHVACTemp

This output network variable is used to source a temperature signal to a control object as a space comfort controller.

1.3.1.2 Configuration Properties

| Configuration | Default Value | Description |
|----------------------|---------------|-----------------------------------|
| SCPTmaxSendTime (49) | 30,0 | Heartbeat maximum send time |
| SCPTminSendTime (52) | 0,5 | Heartbeat minimum send time |
| SCPTovrValue (33) | 22,0 | Temperature value when overwrite |
| SCPTminDeltaTemp () | 0,30 | Temp. change before variable send |
| SCPToffsetTemp () | 0,00 | Temp. hardware offset adjustment |

1.3.1.2.1 SCPTmaxSendTime

This configuration property set the value for the send heartbeat. Maximum period of time that expires before the object automatically transmits the present value of the lux level output NV.

1.3.1.2.2 SCPTminSendTime

This configuration property set the value for the minimum send time. Minimum period between output NV transmissions (maximum transmission rate).

1.3.1.2.3 SCPTovrValue

This configuration property set the value for the output variable, when the object is in override mode. The variable type is a SNVT_temp (). The value is used on the nvoHVACTemp variable from the object.

1.3.1.2.4 SCPTminDeltaTemp

This configuration property is used to configure how large a temperature change, that will activate an event sending out an update on the nvoHVACTemp variable. The value is in

1.3.1.2.5 SCPToffsetTemp

This configuration property type is used the adjust the hardware sensor connected to the comfort and light controller temperature input and compensate the temperature failure on the sensor. The value is subtracted or added from the measured sensor value.

1.3.2 Space Comfort Controller (8500), 1 objects

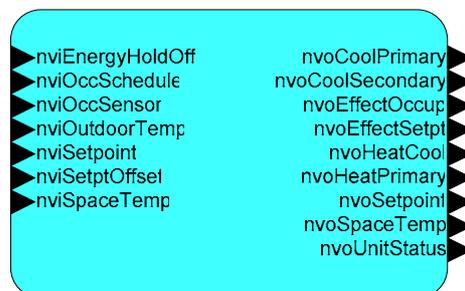
The following section is describing the space comfort controller software object. This object is a temperature controller used to automatic controlling heat and or cool in a zone. The object receive a temperature on its input variable and decide if there is need for heat or cool. The decision is taken with reference to a set of set points configured in the object. Hereafter the controller will use a PI function controlling either heat or cool output.

The space comfort controller has both a primary heat and cool output and a secondary heat and cool output. This gives many possibilities when configuring the controller functions. In cooling situations the ventilation can take the first stage for cooling and a real cooler unit can take the second stage if the first stage cannot cool down, to the set point used.

The controller is designed to work with partition walls, combining two separate controllers into one, when a partition wall is open.

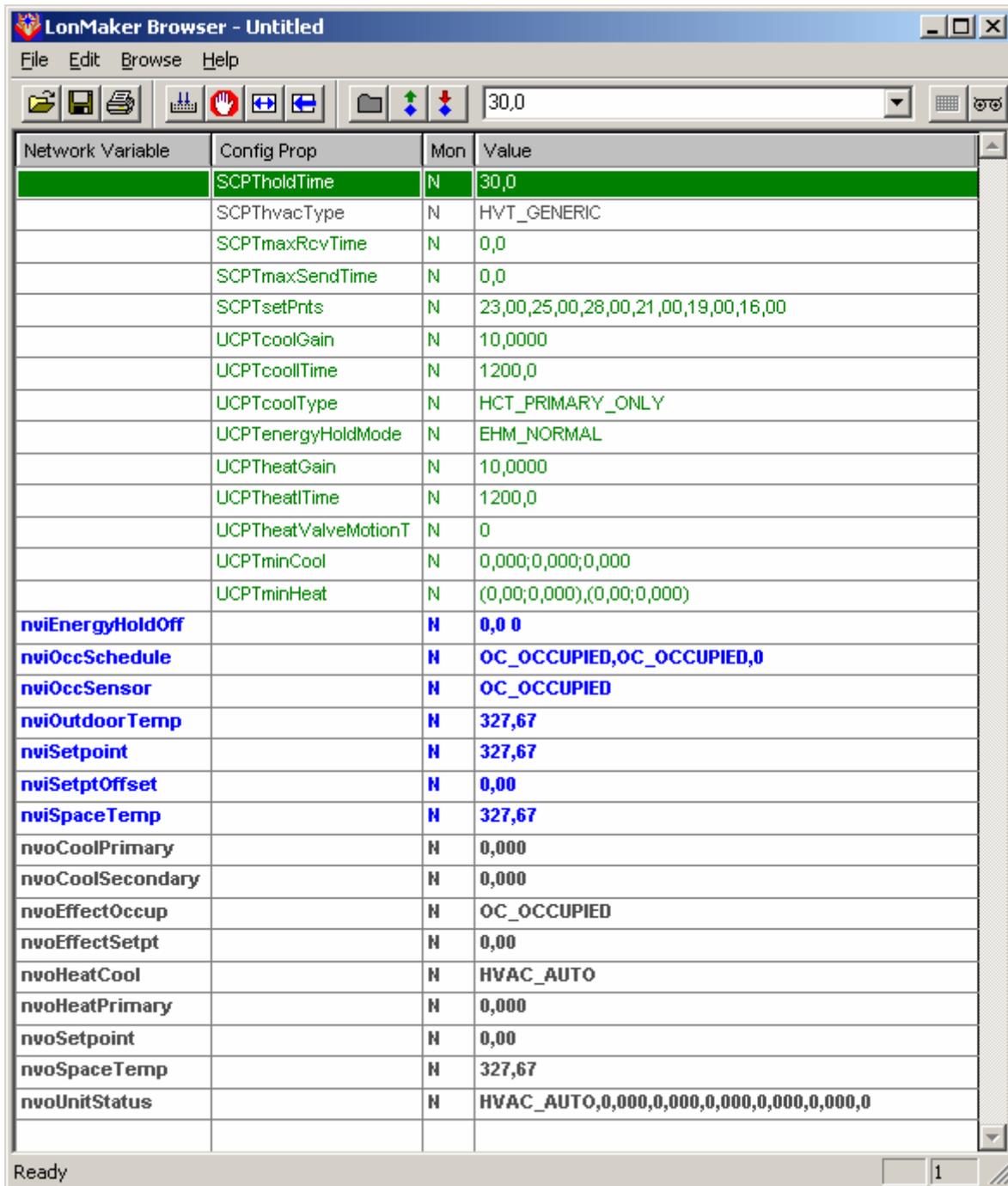
The controller also have build-in compensation when no need for heat in the zone, but it's very cold outside. Using the outdoor temperature compared to a configured minimum level property, a minimum heat signal is send out the heat output even if there is no need for heat in the zone, giving a better space comfort, specially close to the windows area.

| Controller # | Name in LonMaker list |
|--------------|-----------------------|
| 1 | SpaceComfortCtrl |



Space Comfort Controller (SCC) - Generic

Figure 16 The space comfort controller



| Network Variable | Config Prop | Mon | Value |
|-------------------------|----------------------|----------|--|
| | SCPTholdTime | N | 30,0 |
| | SCPTvacType | N | HVT_GENERIC |
| | SCPTmaxRcvTime | N | 0,0 |
| | SCPTmaxSendTime | N | 0,0 |
| | SCPTsetPnts | N | 23,00,25,00,28,00,21,00,19,00,16,00 |
| | UCPTcoolGain | N | 10,0000 |
| | UCPTcoolTime | N | 1200,0 |
| | UCPTcoolType | N | HCT_PRIMARY_ONLY |
| | UCPTenergyHoldMode | N | EHM_NORMAL |
| | UCPTheatGain | N | 10,0000 |
| | UCPTheatTime | N | 1200,0 |
| | UCPTheatValveMotionT | N | 0 |
| | UCPTminCool | N | 0,000;0,000;0,000 |
| | UCPTminHeat | N | (0,00;0,000),(0,00;0,000) |
| nviEnergyHoldOff | | N | 0,0 0 |
| nviOccSchedule | | N | OC_OCCUPIED,OC_OCCUPIED,0 |
| nviOccSensor | | N | OC_OCCUPIED |
| nviOutdoorTemp | | N | 327,67 |
| nviSetpoint | | N | 327,67 |
| nviSetptOffset | | N | 0,00 |
| nviSpaceTemp | | N | 327,67 |
| nvoCoolPrimary | | N | 0,000 |
| nvoCoolSecondary | | N | 0,000 |
| nvoEffectOccup | | N | OC_OCCUPIED |
| nvoEffectSetpt | | N | 0,00 |
| nvoHeatCool | | N | HVAC_AUTO |
| nvoHeatPrimary | | N | 0,000 |
| nvoSetpoint | | N | 0,00 |
| nvoSpaceTemp | | N | 327,67 |
| nvoUnitStatus | | N | HVAC_AUTO,0,000,0,000,0,000,0,000,0,000,0 |

Figure 17 Browsing the space comfort controller object

1.3.2.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|------------------|---|------------------------|---------------------------|
| nviEnergyHoldOff | 0,0 0 | SNVT_switch (95) | Energy hold-off input |
| nviOccSchedule | OC_OCCUPIED, OC_OCCUPIED, 0 | SNVT_tod_event (128) | Operation mode selection |
| nviOccSensor | OC_OCCUPIED | SNVT_occupancy (109) | Occupancy sensor input |
| nviOutdoorTmp | 327,067 | SNVT_temp_p (105) | Outdoor temperature |
| nviSetpoint | 327,067 | SNVT_temp_p (105) | Absolute set point |
| nviSetptOffset | 0,0 | SNVT_temp_p (105) | Set point offset value |
| nviSpaceTemp | 327,067 | SNVT_temp_p (105) | Temperature input |
| | | | |
| nvoCoolPrimary | 0,0 | SNVT_lev_percent (81) | Primary cool out |
| nvoCoolSecondary | 0,0 | SNVT_lev_percent (81) | Secondary cool out |
| nvoEffectOccup | OC_OCCUPIED | SNVT_occupancy (109) | Effective occupancy out |
| nvoEffectSetpt | 0,0 | SNVT_temp_p (105) | Effective setpoint output |
| nvoHeatCool | HVAC_AUTO | SNVT_hvac_mode (108) | Effective heat/cool out |
| nvoHeatPrimary | 0,0 | SNVT_lev_percent (81) | Primary heat out |
| nvoSetpoint | 0,0 | SNVT_temp_p (105) | Local setpoint output |
| nvoSpaceTemp | 327,067 | SNVT_temp_p (105) | Effective temp. output |
| nvoUnitStatus | HVAC_AUTO, 0,000,0,000, 0,000,0,000, 0,000,0 | SNVT_hvac_status (112) | Unit status output |

1.3.2.1.1 nviEnergyHoldOff

Energy hold-off input. This input is used to stop heating and cooling while allowing the unit to protect the space from temperature extremes.

1.3.2.1.2 nviOccSchedule

Occupancy scheduler input. Command the SCC into different occupancy modes using the SNVT_tod_event structure.

1.3.2.1.2.1 SNVT_tod_event structure:

| Field 1 | Field 2 | Field 3 |
|---------------|------------|--------------------|
| current_state | next_state | time_to_next_state |

Tod event field description

| Field | Description | Refference |
|----------------------|--------------------|--------------------|
| 1 current_state | Occupancy, current | Occup_t table |
| 2 next_state | Occupancy, next | Occup_t table |
| 3 time_to_next_state | Time to next state | 0 to 65535 minutes |

1.3.2.1.2.1.1 Occup_t, Occupancy code names table:

| Occup_t | Index |
|---------------|-------|
| OC_NUL | -1 |
| OC_OCCUPIED | 0 |
| OC_UNOCCUPIED | 1 |
| OC_BYPASS | 2 |
| OC_STANDBY | 3 |

1.3.2.1.3 nviOccSensor

Occupancy sensor input. Indicate the presence of occupants in the controlled space/zone. This input start the hold timer and count down from the set value in the hold time property SCPTholdTime. The variable is SNVT_occupancy.

1.3.2.1.4 nviOutdoorTemp

Outdoor air temperature input. Represents information from an outdoor air temperature sensor. Is used to calculate minimum heat, compensation for chilled windows or other building parts, when the outdoor temperature is very low.

1.3.2.1.5 nviSetpoint

Absolute temperature setpoint input. Allow the temperature set points for the occupied and standby modes to be changed via the network directly.

1.3.2.1.6 nviSetptOffstet

Setpoint offset input. Shifts the effective occupied and standby temperature setpoints by adding this value to the present setpoints.

1.3.2.1.7 nviSpaceTemp

Space temperature input. Connect an external space temperature sensor to the object.

1.3.2.1.8 nvoCoolPrimary

Primary cool output variable. This variable is a SNVT_lev_percent and is used to control a cool actuator object.



1.3.2.1.9 nvoCoolSecondary

Secondary cool output variable. This variable is a SNVT_lev_percent and is used to control a cool actuator object.

1.3.2.1.10 nvoEffectOccup

Effective occupancy output. Actual occupancy mode of the unit

1.3.2.1.11 nvoEffectSetpt

Effective setpoint output. Monitors the effective temperature setpoint and send it out on the network.

1.3.2.1.12 nvoHeatCool

Effective heat or cool output. Actual heat or cool mode of the unit.

1.3.2.1.13 nvoHeatPrimary

Primary heat output variable. This variable is a SNVT_lev_percent and is used to control a heat actuator object.

1.3.2.1.14 nvoSetpoint

Local setpoint output. Space temperature setpoint value if a setpoint device is hardwired.

1.3.2.1.15 nvoSpaceTemp

Effective space temperature output. Used to monitor the effective space temperature that the SCC is using for control.

1.3.2.1.16 nvoUnitStatus

Unit status output. Reports the SCC status. This output variable is a structure giving a total status for the space comfort controller:

mode, heat_output_primary, heat_output_secondary, cool_output, econ_output, fan_output, in_alarm

Structure fields

| # 1 | # 2 | # 3 | # 4 | # 5 | # 6 | # 7 |
|------|---------------------|-----------------------|-------------|-------------|------------|----------|
| Mode | heat_output_primary | heat_output_secondary | cool_output | econ_output | fan_output | in_alarm |



Fields descriptions

| Field | Description | Refference |
|-------------------------|-------------------|-----------------|
| 1 mode | HVAC mode names | hvac_t |
| 2 heat_output_primary | Primary heat | % of full scale |
| 3 heat_output_secondary | Secondary heat | % of full scale |
| 4 cool_output | Cool output | % of full scale |
| 5 econ_output | Economizer output | % of full scale |
| 6 fan_output | Fan output | % of full scale |
| 7 in_alarm | In alarm state | alarm value |

hvac_t description table:

| hvac_t | Index # |
|------------------|---------|
| HVAC_NUL | -1 |
| HVAC_AUTO | 0 |
| HVAC_HEAT | 1 |
| HVAC_MRNG_WRMUP | 2 |
| HVAC_COOL | 3 |
| HVAC_NIGHT_PURGE | 4 |
| HVAC_PRE_COOL | 5 |
| HVAC_OFF | 6 |
| HVAC_TEST | 7 |
| HVAC_EMERG_HEAT | 8 |
| HVAC_FAN_ONLY | 9 |
| HVAC_FREE_COOL | 10 |
| HVAC_ICE | 11 |
| HVAC_MAX_HEAT | 12 |
| HVAC_ECONOMY | 13 |
| HVAC_DEHUMID | 14 |
| HVAC_CALIBRATE | 15 |
| HVAC_EMERG_COOL | 16 |
| HVAC_EMERG_STEAM | 17 |

1.3.2.2 Configuration Properties

| Configuration | Default Value | Description |
|----------------------|---|--------------------------------|
| SCPTholdTime | 30,0 | Occupancy timer value |
| SCPTvacType (169) | HVT_GENERIC | HVAC unit type identifier |
| SCPTmaxRcvTime | 250,0 | Heartbeat maximum receive time |
| SCPTmaxSendTime | 120,0 | Heartbeat maximum send time |
| SCPTsetPnts | 23,00,25,00,28,00,21,00, 19,00,16,00 | Set points for heat and cool |
| UCPTcoolGain | 10,0000 | PI gain cool in procent |
| UCPTcoolTime | 1200,0 | PI time cool in seconds |
| UCPTcoolType | HCT_PRIMARY_ONLY | Select cool type function |
| UCPTenergyHoldMode | EHM_NORMAL | Energy hold mode selection |
| UCPTheatGain | 10,0000 | PI gain heat in procent |
| UCPTheatTime | 1200,0 | PI time heat in seconds |
| UCPTheatValveMotionT | 0 | Interval between valve motion |
| UCPTminCool | 0,000;0,000;0,000 | Minimum cool level selection |
| UCPTminHeat | (-10,00;30,000),(10,00;0,000) | Minimum heat level selection |

1.3.2.2.1 SCPTholdTime

This configuration property is used to set the timer value for the occupancy timer. This timer is used to hold the space/zone occupied after the last occupancy input. The timer is described in seconds.

1.3.2.2.2 SCPTvacType

HVAC unit type identifier. This value is set by the manufacturer to allow an integrator to know the function of this SCC device. The default value is HVT_GENERIC.

| SCPTvacType | Index # |
|----------------|---------|
| HVT_NUL | -1 |
| HVT_GENERIC | 0 |
| HVT_FAN_COIL | 1 |
| HVT_VAV | 2 |
| HVT_HEAT_PUMP | 3 |
| HVT_ROOFTOP | 4 |
| HVT_UNIT_VENT | 5 |
| HVT_CHILL_CEIL | 6 |
| HVT_RADIATOR | 7 |
| HVT_AHU | 8 |
| HVT_SELF_CONT | 9 |

1.3.2.2.3 SCPTmaxRcvTime

Maximum receive time. The maximum period of time that may expire with no updates on the associated input network variables before the object goes into heartbeat failure mode. A zero value disables. This configuration property is used for the SNVT input

1.3.2.2.4 SCPTmaxSendTime

Send heartbeat. Maximum period of time that expires before the specified NV outputs will automatically be updated again.

1.3.2.2.5 SCPTsetPnts

This configuration property is a structure used to set the setpoints for heat and cool, in occupied, unoccupied and standby mode.

Default values

| Field 1 | Field 2 | Field 3 | Field 4 | Field 5 | Field 6 |
|---------|---------|---------|---------|---------|---------|
| 23,00 | 25,00 | 28,00 | 21,00 | 19,00 | 16,00 |

Setpoints structure:

| Field 1 | Field 2 | Field 3 | Field 4 | Field 5 | Field 6 |
|---------------|--------------|-----------------|---------------|--------------|-----------------|
| occupied_cool | standby_cool | unoccupied_cool | occupied_heat | standby_heat | unoccupied_heat |

Field description:

| Field # | Description |
|-------------------|--|
| 1 occupied_cool | Occupied cooling setpoint. degrees Celsius |
| 2 standby_cool | Standby cooling setpoint. degrees Celsius |
| 3 unoccupied_cool | Unoccupied cooling setpoint. degrees Celsius |
| 4 occupied_heat | Occupied heating setpoint. degrees Celsius |
| 5 standby_heat | Standby heating setpoint. degrees Celsius |
| 6 unoccupied_heat | Unoccupied heating setpoint. degrees Celsius |

1.3.2.2.6 UCPTcoolGain

This configuration property is used to select the gain value in the PI regulator for the cooling part.

1.3.2.2.7 UCPTcoolITime

This configuration property is used to select the integral time value in the PI regulator for the cooling part.



1.3.2.2.8 UCPTcoolType

Select the function for the cool outputs.

HCT_PRIMARY_ONLY
HCT_PRIMARY_THEN_SECONDARY
HCT_SECONDARY_ONLY

1.3.2.2.9 UCPTenergyHoldMode

This configuration property is used to select the type of energy hold, that is used when the space comfort controller is in energy hold mode.

1.3.2.2.10 UCPTheatGain

This configuration property is used to select the gain value in the PI regulator for the heating part.

1.3.2.2.11 UCPTheatITime

This configuration property is used to select the integral time value in the PI regulator for the heating part.

1.3.2.2.12 UCPTheatValveMotionT

This configuration property is used to select the interval between an automatic valve motion on the heat valves.

1.3.2.2.13 UCPTminCool

1.3.2.2.14 UCPTminHeat

1.3.3 Heat Actuator, 1 objects

The following describes the heat actuator object. The object is used to control a hardware actuator for heat valves. The object is used to control both a digital, analog and floating actuators. The hardware output is either 24Vac ON/OFF or a 0 to 10Vdc signal with fixed 24Vac supply.

| Heat # | Connector | Name in LonMaker list |
|--------|-----------|-----------------------|
| 1 | HEAT | HeatActuator |

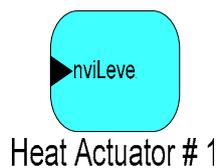


Figure 18 The heat actuator object

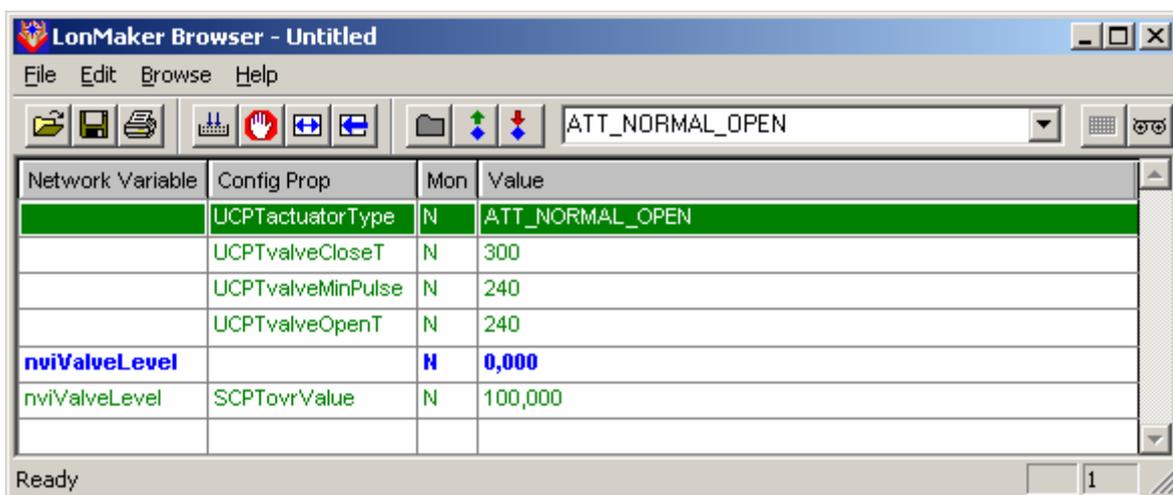


Figure 19 Browsing the heat actuator object

The heat actuator object receive a control level in % and with help from the configuration properties a calculation is made, to decide how to actuate the hardware outputs.

When the control input is under the valve minimum pulse limit level, the algorithm will calculate the pause length between the ON pulses and use a fixed ON length pulse using the valve minimum pulse with a security margin.

When the control input is over the valve minimum pulse limit level, the algorithm will calculate a duty cycle modulation using the valve open, close and minimum time compensating for the slow reaction of the actuator. This will generate an average flow trough the valve, matching the control input % level.

When the object is used as an analog actuator control, the hardware output is modulated to a 0 to 10 Vdc signal related to the control input in %, giving approximately a flow trough the valve matching the control input % level.

1.3.3.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|---------------|---------------|-----------------------|---------------------|
| nviValueLevel | 0 % | SNVT_lev_percent (82) | Input control value |

1.3.3.1.1 nviValueLevel

This input network variable is the control level for the hardware actuator. It will control the actuator level in % of the full actuator range from 0 to 100%.

1.3.3.2 Configuration Properties

| Configuration | Default Value | Description |
|--------------------------|-----------------|-------------------------|
| SCPTovrValue (33) | 100 % | SNVT overwrite value |
| UCPTvalveCloseT (3:43) | 300 sec. | Actuator closing time |
| UCPTvalveMinPulse (3:44) | 240 sec. | Minimum reaction time |
| UCPTvalveOpenT (3:42) | 240 sec. | Actuator opening time |
| UCPTactuatorType | ATT_NORMAL_OPEN | Actuator type selection |

1.3.3.2.1 SCPTovrValue

This configuration property set the level on the hardware output, when the object is in override mode. The variable type is a SNVT_lev_percent (82). The value is used on the hardware output.

1.3.3.2.2 UCPTvalveCloseT

UCPTvalveCloseT is shorted from, valve close time configuration property.

The UCPTvalveCloseT configuration is specifying the speed the actuator can close the valve with. The closing time is used in the algorithm to compensate for the slow actuator speed and make sure that the actuator control the valve best possible. Consult the actuator documentation for getting the correct value.

1.3.3.2.3 UCPTvalveMinPulse

UCPTvalveMinPulse is shorted from, valve minimum pulse length configuration property.

The UCPTvalveMinPulse configuration is specifying the minimum pulse length possible, getting a water flow in the valve, even if the actuator are not opening 100%. If you can find this value, the algorithm can calculate a better control output. This will also give possibility to use less water, when compensating for cold windows surfaces.

1.3.3.2.4 UCPTvalveOpenT

UCPTvalveOpenT is shorted from, valve open time configuration property.



The UCPTvalveOpenT configuration is specifying the speed the actuator can open the valve with. The opening time is used in the algorithm to compensate for the slow actuator speed and make sure that the actuator control the valve best possible. Consult the actuator documentation for getting the correct value

1.3.3.2.5 UCPTactuatorType

UCPTvalveType is shorted from, valve actuator type configuration property.

The following selections are possible:

ATT_NORMAL_OPEN

ATT_NORMAL_CLOSED

1.3.4 Cool Actuator, 1 objects

The following describes the cool actuator object. The object is used to control a hardware actuator for cool valves. The object can be used on both digital, analog and flowing actuators. The hardware output is 0 til 10Vdc and a fixed 24Vdc.

| Cool # | DC # | 0-10 # | Name in LonMaker list |
|--------|----------|--------|-----------------------|
| 1 | COOL +24 | COOL | CoolActuator |

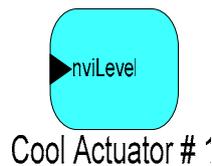


Figure 20 The cool actuator object

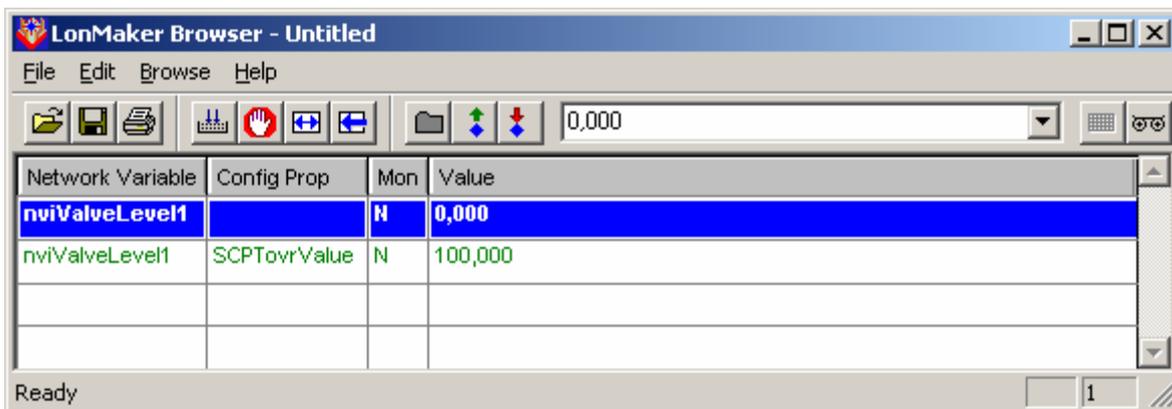


Figure 21 Browsing the cool actuator object

1.3.4.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|----------------|---------------|-----------------------|---------------------|
| nviValueLevel1 | 0 % | SNVT_lev_percent (82) | Input control value |

1.3.4.1.1 nviValueLevel1

This input network variable is the control level for the hardware actuator. It will control the actuator level in % of the full actuator range from 0 to 100%.



1.3.4.2 Configuration Properties

| Configuration | Default Value | Description |
|-------------------|---------------|----------------------|
| SCPTovrValue (33) | 100 % | SNVT overwrite value |

1.3.4.2.1 SCPTovrValue

This configuration property set the level on the hardware output, when the object is in override mode. The variable type is a SNVT_lev_percent (82). The value is used on the hardware output.

1.3.5 Ventilation Actuator CAV, 1 objects

The following describes the ventilation actuator object. The object is used to control a hardware actuator for ventilation actuators. The hardware output is 24Vdc ON/OFF.

| CAV # | DC # | Name in LonMaker list |
|-------|------|-----------------------|
| 1 | CAV | CavActuator |



CAV Actuator # 1

Figure 22 The CAV actuator object

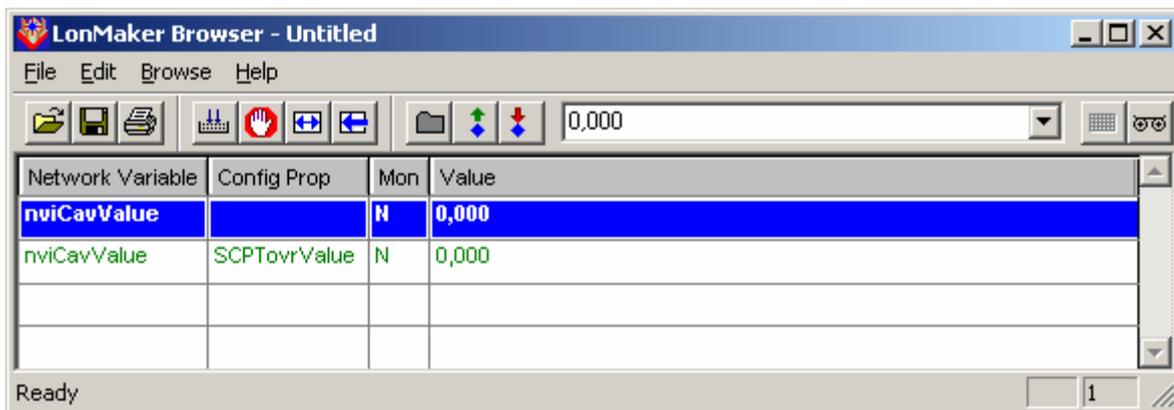


Figure 23 Browsing the CAV actuator object

The ventilation actuator object receive a control level in % and with help from the configuration properties a calculation is made, to decide how to actuate the hardware outputs.

When the object is used as a digital actuator control, the hardware output is turned ON and OFF with an interval related to the control level in %, giving approximately an average flow trough the valve matching the control input % level.



1.3.5.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|---------------|---------------|-----------------------|---------------------|
| nviValueLevel | 0 % | SNVT_lev_percent (82) | Input control value |

1.3.5.1.1 nviValueLevel

This input network variable is the control level for the hardware actuator. It will control the actuator level in % of the full actuator range from 0 to 100%.

1.3.5.2 Configuration Properties

| Configuration | Default Value | Description |
|-------------------|---------------|----------------------|
| SCPTovrValue (33) | 100 % | SNVT overwrite value |

1.3.5.2.1 SCPTovrValue

This configuration property set the level on the hardware output, when the object is in override mode. The variable type is a SNVT_lev_percent (82). The value is used on the hardware output.

1.3.6 Ventilation Actuator VAV, 1 objects

The following describes the ventilation actuator object. The object is used to control a hardware actuator for ventilation actuators. The hardware output is 0 til 10Vdc and a fixed 24Vdc.

| Cool # | DC # | 0-10 # | Name in LonMaker list |
|--------|---------|--------|-----------------------|
| 1 | VAV +24 | VAV | |

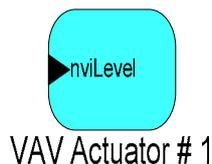


Figure 24 The VAV actuator object

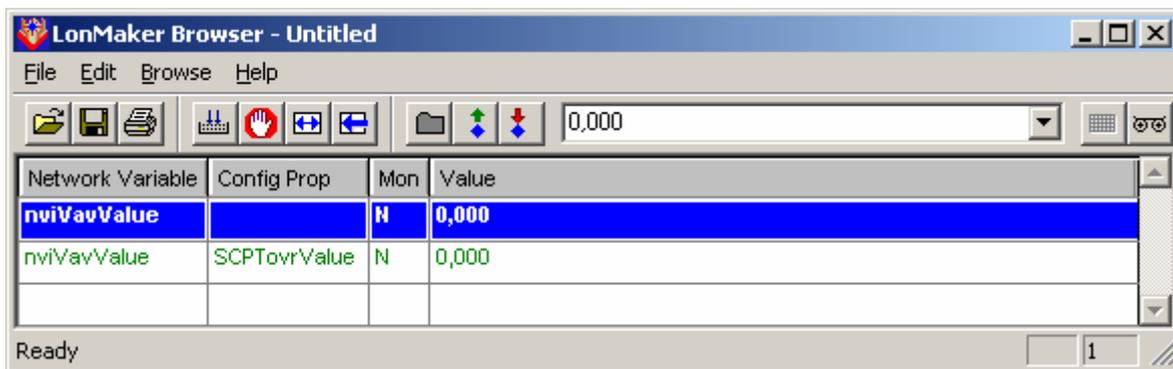


Figure 25 Browsing the VAV actuator object

The ventilation actuator object receive a control level in % and with help from the configuration properties a calculation is made, to decide how to actuate the hardware outputs.

The object is used as an analog actuator control, the hardware output is modulated to a 0 to 10 Vdc signal related to the control input in %, giving approximately a flow trough the valve matching the control input % level.

1.3.6.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|---------------|---------------|-----------------------|---------------------|
| nviValueLevel | 0 % | SNVT_lev_percent (82) | Input control value |

1.3.6.1.1 nviValueLevel

This input network variable is the control level for the hardware actuator. It will control the actuator level in % of the full actuator range from 0 to 100%.



1.3.6.2 Configuration Properties

| Configuration | Default Value | Description |
|-------------------|---------------|----------------------|
| SCPTovrValue (33) | 100 % | SNVT overwrite value |

1.3.6.2.1 SCPTovrValue

This configuration property set the level on the hardware output, when the object is in override mode. The variable type is a SNVT_lev_percent (82). The value is used on the hardware output.

1.4 Sunblind profiles:

The following list is describing the sunblind software functions implemented in the Comfort and light controller Lonbox PZM2114.

1.4.1 Sunblind Actuator, 2 objects Not implementet in version 3.00

The sunblind actuator object is controlling the hardware output for the sunblind actuator.

Please consult the hardware manual for more details about AC or DC actuators.



Prolon Sunblind Actuator

Figure 26 The sunblind actuator object



Figure 27 Browsing the sunblind actuator object

1.4.1.1 Network Variable Interface

| Variable name | Default Value | Type | Description |
|-----------------|------------------|--------------------|-----------------------|
| nviSblndSetting | SET_OFF 0,0 0,00 | SNVT_setting (117) | Input setting control |
| nvManOverride | SET_OFF 0,0 0,00 | SNVT_setting (117) | Manuel control |

1.4.1.1.1 nviSblndSetting

Control input for the sunblind actuator. The variable is a SNVT_setting, this is at structure.

The structure:

(function, setting, rotation)



1.4.1.1.2 nvManOverride

Manuel input for the sunblind actuator. The variable is a SNVT_setting, this is at structure.

The structure:
(function, setting, rotation)

1.4.1.2 Configuration Properties

| Configuration | Default Value | Description |
|----------------------|----------------------|--------------------|
| UCPTdriveTime | 150 | Drive time close |

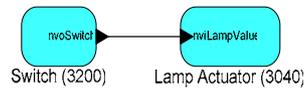
1.4.1.2.1 UCPTdriveTime

The UCPTdriveTime configuration is specifying the time the actuator drive from start to the end. Consult the actuator documentation for getting the correct value. The value is in seconds.

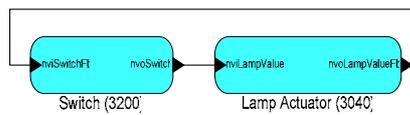
2 Applications and bindings

2.1 Manual Control, Light Bindings

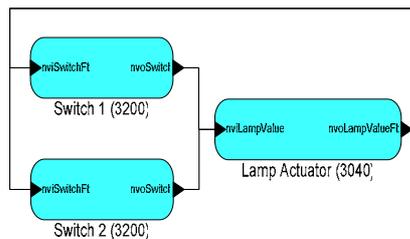
2.1.1 Simple on and off.



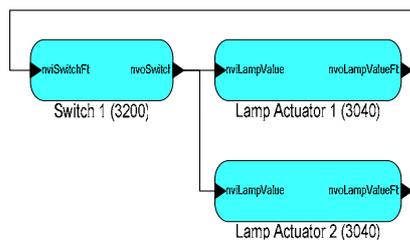
2.1.2 Simple on and off with feed back.



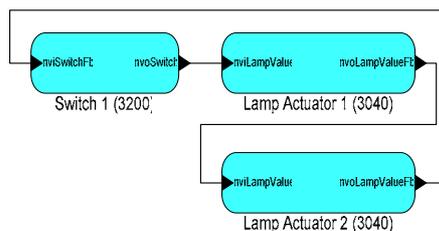
2.1.3 Multi on and off with feed back.



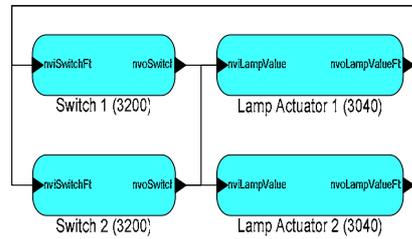
2.1.4 Simple on and off with multi actuators 1.



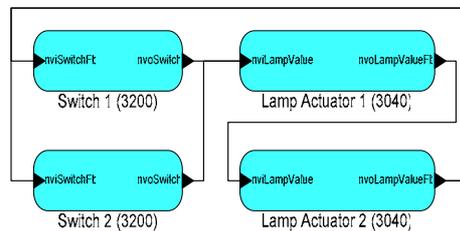
2.1.5 Simple on and off with multi actuators 2.



2.1.6 Multi on and off with multi actuators 1.

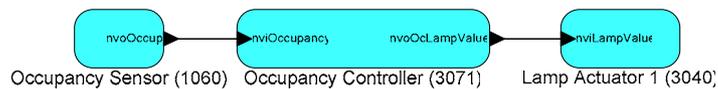


2.1.7 Multi on and off with multi actuators 2.

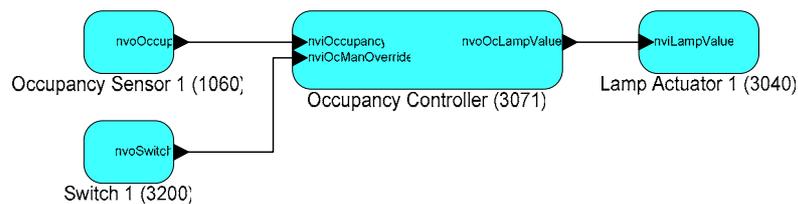


2.2 Occupancy automatic, Light Bindings

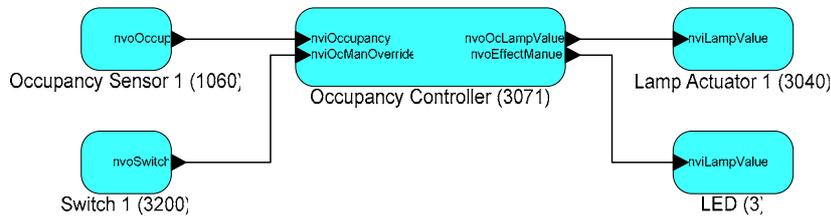
2.2.1 Occupancy on off.



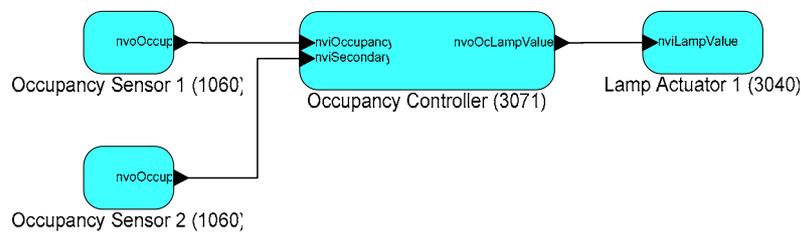
2.2.2 Occupancy on off, with manual override.



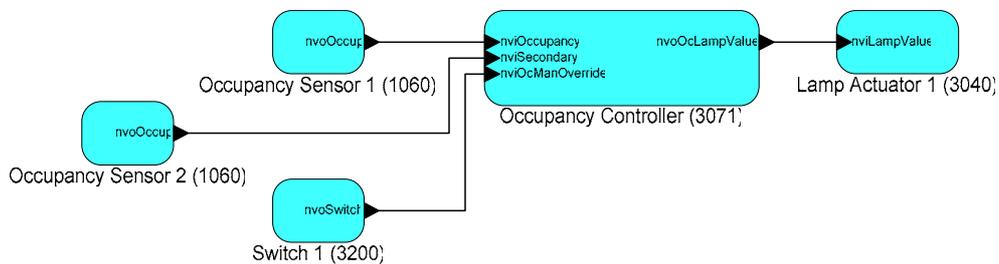
2.2.3 Occupancy on off, manual override and manual control LED.



2.2.4 Occupancy and neighbour occupancy on off.

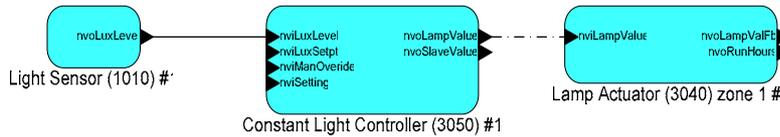


2.2.5 Occupancy, neighbour occupancy on off and manual override.

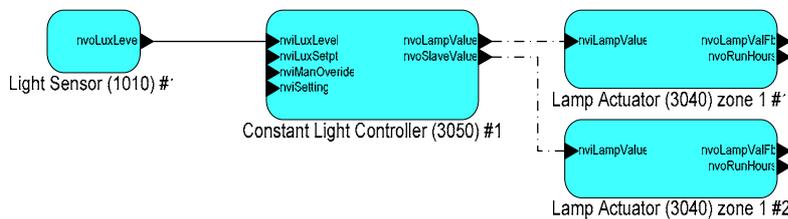


2.3 Constant light automatic, Light Bindings

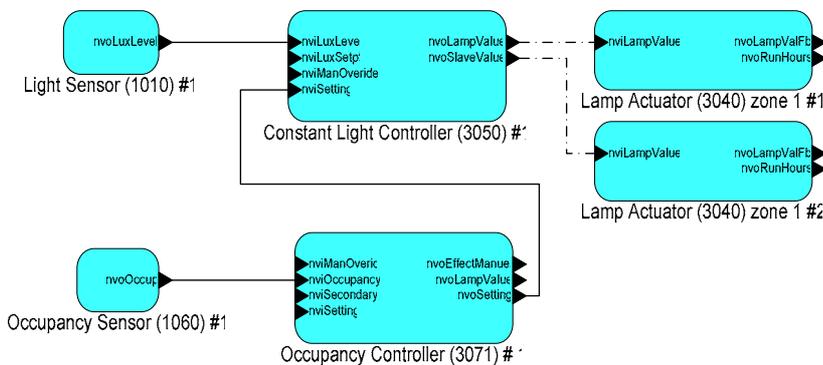
2.3.1 Constant light basis automatic



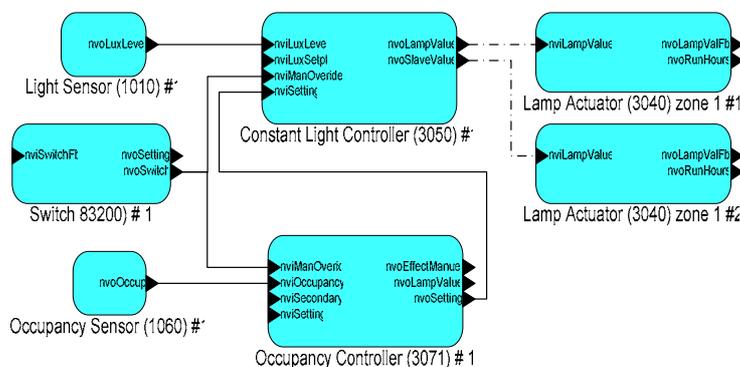
2.3.2 Constant light with slave output



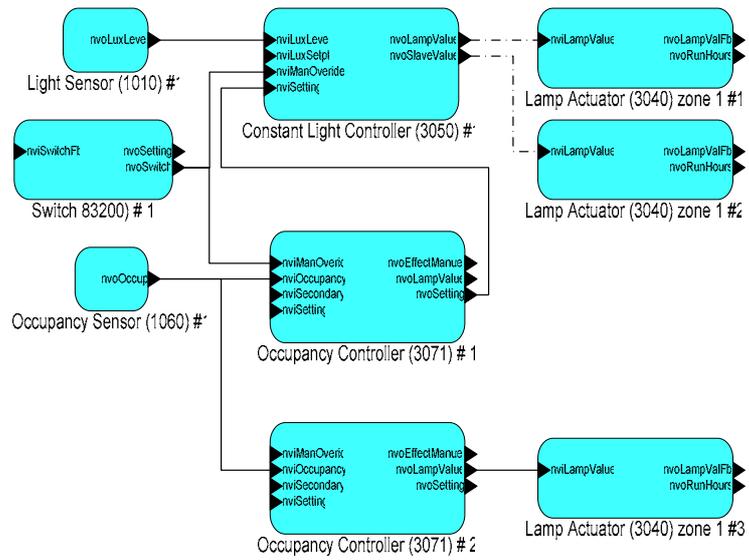
2.3.3 Constant light and occupancy control



2.3.4 Constant light, occupancy and manual control

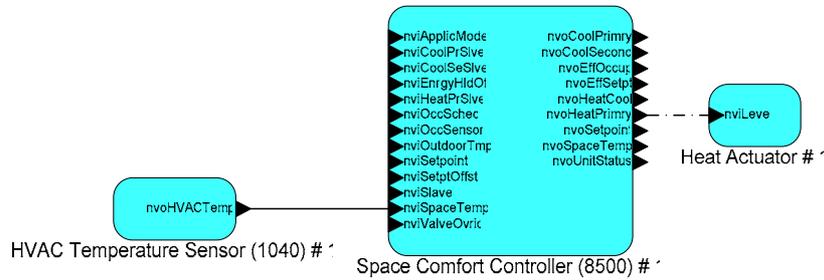


2.3.5 Constant light, dual occupancy and a manual control

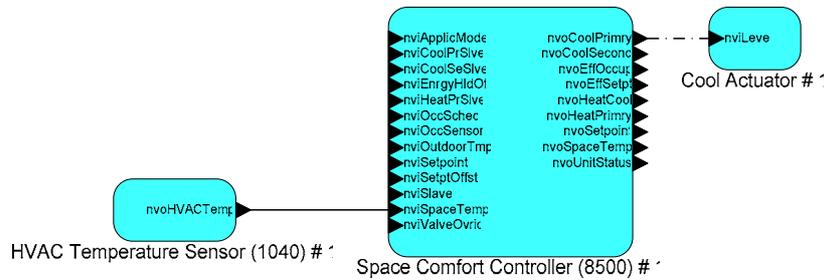


2.4 Space Comfort

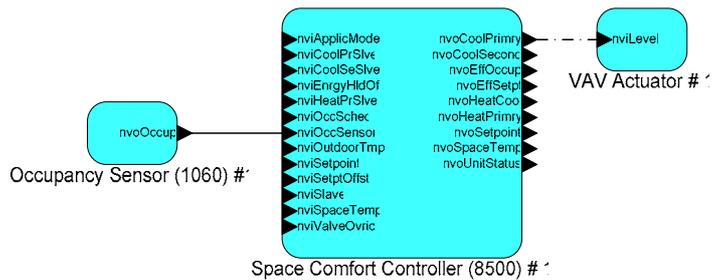
2.4.1 Heating



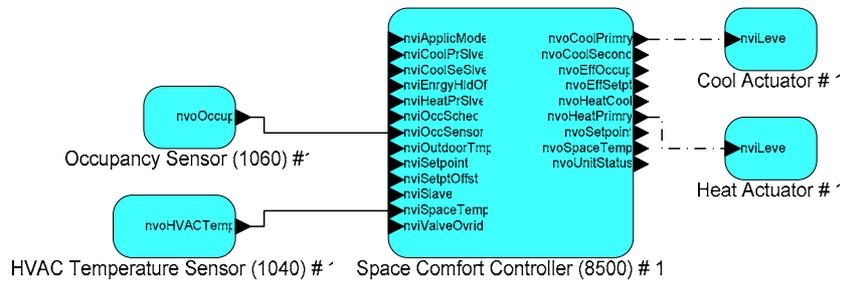
2.4.2 Cooling



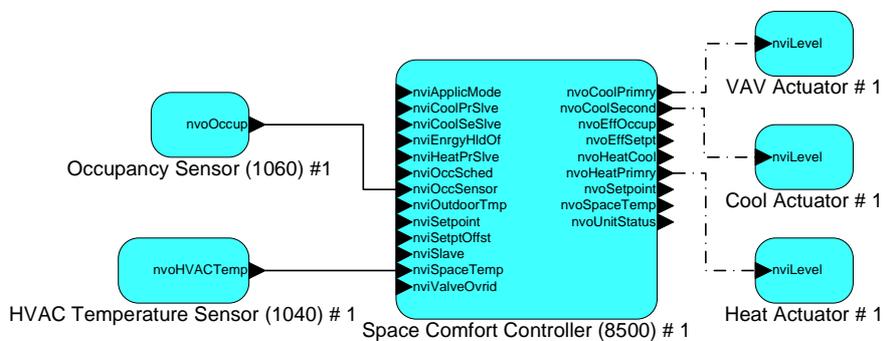
2.4.3 Ventilation with occupancy control



2.4.4 Heat and cool with occupancy automatic



2.4.5 Heat, cool and ventilation with occupancy automatic





3 Figur list

| | |
|---|----|
| Figure 1 The occupancy sensor object | 5 |
| Figure 2 Browsing the occupancy sensor object | 5 |
| Figure 3 The Occupancy controller object | 8 |
| Figure 4 Browsing the occupancy controller object | 9 |
| Figure 5 The light sensor object | 12 |
| Figure 6 Browsing the light sensor object | 12 |
| Figure 7 The lamp actuator object | 15 |
| Figure 8 Browsing the lamp actuator object | 15 |
| Figure 9 Typically use of the lamp actuator object | 18 |
| Figure 10 The constant light controller object | 19 |
| Figure 11 Browsing the constant light controller object | 20 |
| Figure 12 Understanding the light slave output | 22 |
| Figure 13 Constant light controller typically use | 24 |
| Figure 14 The temperature sensor object | 25 |
| Figure 15 Browsing the temperature sensor object | 25 |
| Figure 16 The space comfort controller | 27 |
| Figure 17 Browsing the space comfort controller object | 28 |
| Figure 18 The heat actuator object | 36 |
| Figure 19 Browsing the heat actuator object | 36 |
| Figure 20 The cool actuator object | 39 |
| Figure 21 Browsing the cool actuator object | 39 |
| Figure 22 The CAV actuator object | 41 |
| Figure 23 Browsing the CAV actuator object | 41 |
| Figure 24 The VAV actuator object | 43 |
| Figure 25 Browsing the VAV actuator object | 43 |
| Figure 26 The sunblind actuator object | 45 |
| Figure 27 Browsing the sunblind actuator object | 45 |