Version 1.0
December 2013
Luminair Controller: 35.14

LONMARK®
Smart Luminaire Controller

Overview

This document describes the Functional Profile of a Luminaire Controller Object designed to support broadcast scheduling, and peer-connected traffic/occupancy signals.

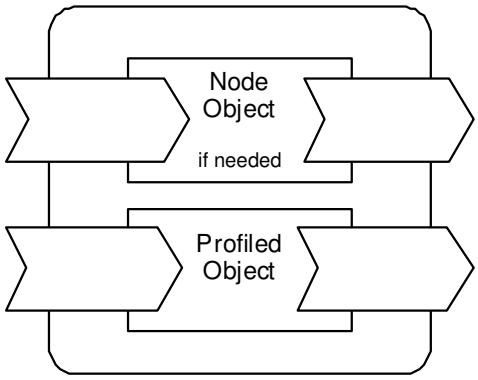

Figure 1 Node Concept

Example Usage

This Profile is designed for lighting applications over using power line communications. It is designed to optimize resource use both on the processor and the segment controller. This profile is designed to fit in devices with limited memory resources.

Object Details

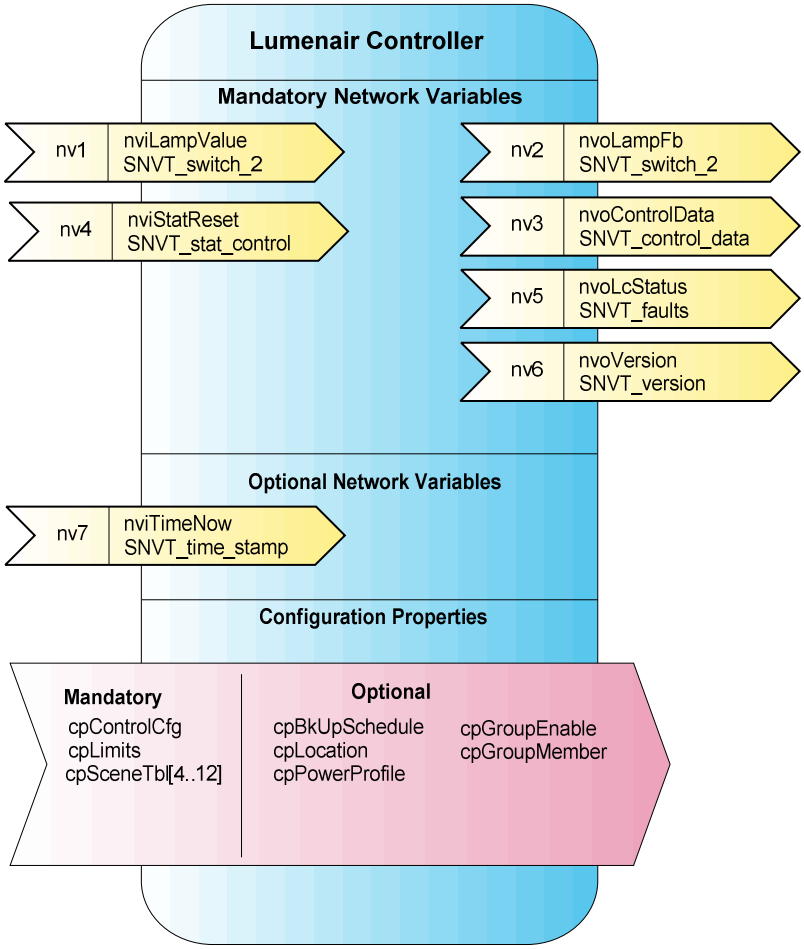

Figure 2 Object Details

Table 1 SNVT Details

NV # (M/O)*	Variable Name	SNVT Name	SNVT Index	Description
1 (M)	nviLampValue	SNVT_switch_2	189	Used for scheduled and Occupancy events.
2 (M)	nviStatReset	SNVT_stat_control	216	Sets/initializes energy, runtime, and error counts.
3 (M)	nvoLampFb	SNVT_switch_2	189	Feedback of current nviLampValue.
4(M)	nvoControlData	SNVT_control_data	218	A structured variable describing all current operating values and state of the SLC.
5(M)	nvoLcStatus	SNVT_faults	217	Latched alarm values are updated only when the condition is asserted active to limit alarm log size and minimize the data sent to the system management software.
6(M)	nvoVersion	SNVT_version	220	A structure with three fields `major.minor.build`.
7(O)	nviTimeNow	SNVT_time_stamp	84	For LCs supporting a HW real-time clock, updates to this variable set the clock time. The profile implementation should display the local time value by updating the value allowing a UI to poll the value to determine the local time base.

* M = mandatory, O = optional

Table 2 SCPT Details

Man. Opt. *	SCPT Name NV Name Type or SNVT	SCPT Index	Associated NVs **	Description
Man	SCPTcontrolCfg cpControlCfg SNVT_control_cfg	382	Entire Object	Defines many of the operating parameters for the SLC.
Man	SCPTlimits cpLimits SNVT_fault_limits	383	Entire Object	Used for alarm thresholds as defined in Alarming, below.
Man	SCPTsceneDef cpSceneTbl structure	384	Entire Object	Defines a table containing between 4 and 12 lighting scenes. Each scene is defined by a scene_number, a level, an unoccupied_scene_number which isused when the controller determines the luminaire light levels should be lowered due to lack of traffic
Opt	SCPTgeoLocatoin cpLocation SNVT_geo_loc	350	Entire Object	Provides tagging for GPS location, and physical asset tagging. Version 13.04 standard type.
Opt	SCPTlightingGroupMembership cpGroupMember structure	361	nviLampValue	Defines membership in one or more lighting groups
Opt	SCPTbkupSchedule cpBkUpSchedule enumeration structure	344	Entire Object	Defines a backup schedule to be used if the device determines the segment controller is not updating the one of the control inputs. Using this Cp requires the node to apply only if it has a valid local time value.
Opt	SCPTpowerProfile cpPowerProfile SNVT_power	381	Entire Object	Defines the nominal power measured at 5 commanded nviLampValues (.5% 25% 50% 75% 100%)

* Man = mandatory, Opt = optional

** List of NVs to which this configuration property applies.

Mandatory Network Variables

Lamp Value Input

```
network input sd_string("@p|1") SNVT_switch_2
nviLampValue;
```

This input is used to recall control levels (using `.state = SW_RECALL_SCENE`) and to process SW_SET_OCCUPIED and SW_SET_UNOCCUPIED signals originating from peer sensor devices. The control levels defined by the scene table define the range of control from 0-100% which maps to control voltages determined by minControlV and maxControlV defined in cpControlCfg defined later in the document. A fixture used near an intersection, for example, may have a maxControlV of 10V to generate full light output, while the same fixture applied to a residential street may clip the light level maximum output by setting maxControlV to 9.1V. The value setting is a %, not a delivered light level. The response of a driver/ballast is implementation dependent over the range of 0-100% level.

A detailed discussion on how the SLC responds to updates coming from a scheduling device, and multiple traffic/occupancy sensors is described in the *Additional Considerations* section of this document.

Valid Range

The valid Range of SNVT_switch_2. The SLC must handle updates that use `.state == (SW_RECALL_SCENE, SW_SET_OCCUPIED, SW_SET_UNOCCUPIED)`. Temporary overrides are implemented by driving the control level according to the level defined when the .state == SW_SET_LEVEL for two update intervals recorded for .state == SW_RECALL_SCENE. Supporting response to of `.state == (SW_SET_OFF, SW_SET_ON, SW_SET_LEVEL)` are optional.

Default Value

None specified.

Configuration Considerations

Note that `nviLampValue.state` is set to `SW_NUL` (-1) at power-up/reset. This provides an indication that the SLC has not seen a control by the segment controller after power up.

Stat Control Input

```
network input sd_string("@p|2") SNVT_stat_control
nviStatReset;
```

This input will set or initializes energy, runtime, and error counts. If the SLC does not support local energy measurement hardware, the SLC will take action on clearing the energy accumulator.

Structure
nviStatReset.cmd (statManage)

An enumeration used to manage various persistent values and performance counters maintained by the controller.

Member	Description
SM_NONE (0)	Take no action
SM_SAVE (1)	Force a write of accumulators that are stored in EEPROM memory such as energy, runtime, and cycle count.
SM_CLEAR_ALL (2)	Zeros all EEPROM store values.
SM_SET_ALL (3)	Allows setting the value of EEPROM managed values to support controller replacement if it is desired to track the values for the attach driver/fixture
SM_CLEAR_VOLATILE (4)	Clears application monitored communication stats as alarm flags
SM_SET_ENERGY (5)	Set only the energy accumulator (see nvoControlData)
SM_SET_RUNTIME (6)	Set only the runtime accumulator (see nvoControlData)
SM_SET_CYCLES (7)	Set only the number of switch cycles count. (see nvoControlData)
SM_CLEAR_ALARMS (8)	Clear the alarm flags now.
SM_CLEAR_COMM_STATS (9)	Clears the application managed comm stats reported in `nvoControlData.nvUpdates`, and `.rcvTimeouts`
SM_CMD_ACK (10)	The SLC will set to command to this value once an action from a command above has been executed. This provides a mean of feedback.

nviStatReset.lockCode (unsigned long)

This is a vender defined code that must be set for a command action to occur. The lock code is used prevent inadvertent actions. The SLC clears this value on update. This is not intended for security, just a measure of protection against unintended clearing of data.

nviStatReset.energySet (SNVT_elec_kwh_l)

Used only when the cmd value is SM_SET_ENERGY and lockCode is set to 8080 to initialize the energy accumulator.

nviStatReset.runtimeSet (SNVT_time_min_p)

Used only when the cmd value is SM_SET_RUNTIME and lockCode is set to 8080 to initialize the runtime accumulator.

nviStatReset.cycleCountSet (SNVT_count)

Used only when the cmd value is set to SM_SET_CYCLES and lockCode is set to 8080 to initialize the cycle count accumulator.

All actions should be qualified against the lockCode before action is taken. The SLC should set the lockCode to 0 after responding to the command.

Valid Range

The valid Range of SNVT_stat_control.

Default Value

None specified.

Configuration Considerations

None specified.

Lamp Feedback Value Output

```
network output sd_string("@p|3") bind_info(any)
SNVT_switch_2 nvoLampFb;
```

This output returns the feedback of the current controlled level derived from nviLampValue. In normal operating conditions, this value will reflect the nviLampValue reporting the current scheduled scene. SW_SET_OCCUPIED and SW_SET_UNOCCUPIED states values are not reflected in this feedback variable, but the effect on the control level is reported in the setting field. The level field should reflect the controlled level calculated from the combined SW_RECALL_SCENE, and Occupancy state understood by the SLC. It is not just a reflection of the nviLampValue. The `.scene_number` field will reflect the scene schedule by the segment controller.

Valid Range

The valid range of SNVT_switch_2. The `state` field will is set to SW_RECALL_SCENE when the scheduler applies a scene request. The `setting` field reports the controlled or measured output level. The `scene_number` field reports the scheduled scene.

Default Value

None specified.

Configuration Considerations

None specified.

When Transmitted

Polled.

Default Service Type

None specified.

SLC Operating Data Output

```
network output sd_string("@p|4") bind_info(any)
SNVT_control_data nvoControlData;
```

This network variable reflects all of this information in one single-structured data type defined by UNVT_control_data.

nvoControlData.power (SNVT_power)

Reflects the instantaneous power consumed by the SLC and controlled fixture. (0.1w resolution) This value is updated each second by the power measurement chip on the SLC. Set to 0xFFFF if not supported.

nvoControlData.energy (SNVT_elec_kwh_l)

Reports the accumulated energy usage using 0.1kwh resolution. The SLC writes this to EEPROM memory every 12 hours, and at the transition to OFF. Installations which kill power to the streetlight segment at dawn must delay the switching of the power for several minutes after the lights are scheduled OFF to allow the controller to store this value in EEPROM memory. Set to 0xFFFFFFFF if not supported)

nvoControlData.runtime (SNVT_time_min_p)

Report the number of operating minutes for the luminaire. The value is stored in EEPROM with each OFF transition. The presentation format for this field is an integer value of hours.

nvoControlData.supplyVoltage (SNVT_volt)

Measured supply voltage using 0.1V resolution. This value is updated every second. Set to 0xFFFF if not supported.

nvoControlData.supplyCurrent (SNVT_amp_ac_mil)

Current supplied by the driver to the lamp measured using 0.001 amp resolution. This value is updated every second. Set to 0xFFFF if not supported.

nvoControlData.cycleCount (SNVT_count)

Number of operating cycles (ON-OFF). This field is updated with each transition to OFF.

nvoControlData.levelFB (SNVT_lev_cont)

0.5% resolution 0-100%. This field tracks the .value field of nvoLampFb. 0% if the state is 0. If the SLC supports power measurement, this could report the level as measured relative to the configured power profile.

nvoControlData.faults (SNVT_faults)

Fault bits. The details are provided in the alarms section. These bits represent current conditions of the last alarm evaluation and not the latched values as reflected in nvoLcStatus.

nvoControlData.nvUpdates (unsigned short)

This is an optional field that is used to assess application level communication performance. Every 30 minutes, this field is updated to report the number of times nviLampValue was updated with the `state` field set to `SW_RECALL_SCENE` in the previous interval. If the defined heartbeat for nviLampValue is defined as 10 minutes, this value reports a value of 2-4 during steady state operation.

nvoControlData.rcvTimeouts (SNVT_count)

The maximum receive timeout for this device (part of cpControlCfg) is set to three times the control input heartbeat. In practice, this number increases only when an update is not received after three heartbeat intervals. RcvTimeout checking is only against updates where the `state` field is `SW_RECALL_SCENE`.

nvoControlData.powerFactor (SNVT_pwr_fact)

SLCs supporting power measurement chips report the measured power factor for the SLCfixture combination. When the controlled load is OFF, this value may be very low (around .3). Alarms against power factor are only evaluated when the load is turned ON. For efficient light operation, it is good practice to limit the control signal the SLC drives to keep the power factor above 0.8. Set to 0xFFFF if not supported.

nvoControlData.LCtemperature (SNVT_temp_p)

SLCs with supporting hardware use this field to report the temperature sensed by the power measurement chip on the SLC. Typical accuracy is +/-5 degrees C. Set to 0xFFFF if not supported.

nvoControlData.LCstate (LC_State)

Reports the current state of the SLC controller. Valid values include: LC_INIT, LC_COOLDOWN, LC_WARMUP, LC_ON, LC_UNOCCUPIED, and LC_OFF.

nvoControlData.driveCurrent (SNVT_amp_ac_mil)

Reports the current measured to drive the light source. Set to 0xFFFF if not supported,

nvoControlData.driveVoltage (SNVT_volt)

Reports the voltage used to drive the light source. Set to 0xFFFF if not supported.

Valid Range

The valid Range of SNVT_control_data

Default Value

None specified.

Configuration Considerations

None specified.

When Transmitted

Polled.

Default Service Type

None specified.

Lamp-Controller Status Output

```
network output sd_string("@p|5") bind_info(any)
SNVT_faults nvoLcStatus;
```

This output is updated on change to minimize the data requirements on the uploading alarm log events. The frequency of changes to this variable is

carefully managed to conserve bytes consumed by data logs. Each alarm flag is latched during the operational ON time. It is cleared at a configured time after the lamp is switched OFF after providing enough time for log delivery. The controller may provide various filters to allow for temporary conditions to be ignored. Fields related to power measurement are valid only on SLCs which include supporting measurement hardware. If the supply voltage alarms are set to be at 30% of nominal, and the supply voltage is up and down continuously, the nvoLcStatus will record on LowVoltage alarm during the operating period. Trying to avoid alarm saturation, and excessive data transmission costs.

Fault Bit Reported by nvoLCstatus are:

nvoLcStatus.LowPower

Measured power is cpLimits.powerLowFault % below the expected power

nvoLcStatus.HighPower

Measured power is cpLimits.powerHighFault % above the expected power

nvoLcStatus.LowSupplyVoltage

Measured voltage is cpLimits.voltageLow % below the voltage defined by cpControlCfg.supplyVoltage

nvoLcStatus.HighSupplyVoltage

Measured voltage is cpLimits.voltageHigh % above the voltage defined by cpControlCfg.supplyVoltage

nvoLcStatus.RelayFailed

Power measured when the load switch relay is disengaged above an implementation specific hard coded value. This would occur if the relay contacts were to weld shut.

nvoLcStatus.FailedStart

Available for controllers that can detect a driver start failure.

nvoLcStatus.Cycling

Used to report a condition where the ballast is cycling the lamp on and off for some reason. May not be used by all implementations

nvoLcStatus.CommMargin

Set if no update to nviLampValue is received before cpLimits.rcvHb timer expires. It is recommended that the rcvHb parameter be set to a value that is 50% longer than configured heartbeat rate for nviLampValue by the segment controller.

nvoLcStatus.RcvTmo

Set if the cpControlCfg.maxRcvTm time expires before a nviLampValue with a .state == SW_RECALL_SCENE is received. This alarm typically results in the load under control being driven to the level defined by cpControlCfg.defaulLev.

nvoLcStatus.HighTemp

Set when the onboard temperature sensor exceeds the operating limit defined by cpLimits.highTemp. .

nvoLcStatus.LampFailed

Occurs when the power measured is below the % low value defined by cpLimits.lampFailFault. Some implementations may have hardware that can sense a lamp failure condition.

nvoLcStatus.LowPF

Set if the power factor of the controlled load falls below cpLimits.pfLow when the load is ON. It is normal for the power factor reported in nvoControlData to be in the range of 0.30 - 0.40 when the load is OFF so this alarm is only set if the load is ON.

The following flags are provided to support device specific faults. The nvoControlData structure is at 31 bytes, so this is an attempt to keep the alarm bits within a 16 bit portion of this network variable.

nvoLcStatus.Mfg1

Meaning determined by the device manufacturer. It is recommended this be used for lowLampVoltage, or driverFailure reporting.

nvoLcStatus.Mfg2

Meaning determined by the device manufacturer. It is recommended this be used for highLampVoltage, or driverTempFailure reporting.

nvoLcStatus.Mfg3

Meaning determined by the device manufacturer. It is recommended this be used for lowLampCurrent, or driver/ballast Failure reporting.

nvoLcStatus.Mfg4

Meaning determined by the device manufacturer. It is recommended this be used for highLampCurrent, or driverCommFailure reporting.

Valid Range

The valid range of SNVT_faults.

Configuration Considerations

None specified

When Transmitted

The output variable is transmitted:

• Upon node reset, after obtaining valid data.

• When the ‘state’ has changed.

Default Service Type

None specified.

SLC Application Version Output

```
network output sd_string("@p|6") bind_info(any)
SNVT_version nvoVersion;
```

This output provides a structure describing major, minor, and build numbers for the SLC implementation.

Valid Range

The valid range of SNVT_version.

Default Value

None specified.

Configuration Considerations

None specified

When Transmitted

Polled.

Default Service Type

None specified.

Optional Network Variables

Initial Time Set Input

```
network input sd_string("@p|7") SNVT_time_stamp
nviTimeNow;
```

This input will set the time. For LCs supporting a HW or SW real-time clock, updates to this variable set the clock time. An implementation using a SW real-time clock must consider the undefined time condition which might exist in the event of a soft or hard reset experienced by the device.

Valid Range

The valid Range of SNVT_time_stamp.

Default Value

None specified.

Configuration Considerations

None specified.

Configuration Properties

Control Configuration (Mandatory)

```
network input config sd_string("&1,p,0\x80,382")
SCPTcontrolCfgcpControlCfg;
```

This configuration property defines many of the operating parameters for the SLC.

The following fields are defined:

cpControlCfg.defaultLev (SNVT_lev_cont)

This is the initial value used before an update to nviLampValue is received by the SLC to drive the lamp value at power ON or reset. This value only applies after the SLC is commissioned by the SmartServer. When unconfigured, the SLC will turn ON the controlled light to full

ON. When power is applied to the SLC, the application enforces a 10s minimum time (even if CoolDownTm =0) before applying this value. A non-zero CoolDownTm will extend this time as required when controlling certain lamp types such as HPS.

The default value is 100%.

cpControlCfg.rampTm (SNVT_time_sec)

Controls how the SLC ramps between level transitions. Only used after the lamp is ON to go between intermediate steps. The SLC limits this value to a maximum of 30s.

The default value is1.5s

cpControlCfg.supplyVoltage (SNVT_volt)

This is the expected nominal supply voltage for the operating fixture. Used of voltage level alarms. The default value is 230V.

cpControlCfg.warmupTm (unsigned short)

The number of minutes the SLC allows the lamp to warm up before allowing dimming commands. During warm up, the dimming commands are deferred. If the SLC is set to go to 75% ON, the SLC will not request the lower power setting for warmupTm minutes before issuing the appropriate dimming level. In LED applications, this is typically 0. Any nviLampValue less than 100% will be delayed while the SLC is in the state LC_WARMUP. This also delays power alarm processing which is important in the case of control of magnetic ballast technology.

The default value is 0 minutes.

cpControlCfg.coolDownTm (unsigned short)

The number of minutes the SLC will delay commands to turn ON after the fixture has been turned OFF. This is important for improving certain lamp technology life times. The SLC enforces a 10s COOLDOWN to allow recovery of the inrush protection circuit. This 10s minimum is subject to change in the future.

The default value is 0 minute.

cpControlCfg.maxRcvTm (SNVT_time_sec)

If the SLC fails to receive an update to nviLampValue for this time the SLC will drive the lamp to the defaultLevel. The segment controller should update nviLampValue up to three times within this period. If the maxRcvTm is 900s, the heartbeat rate of 300s should be used by the segment controller. Note that maxRcvTm = 0 means lights will retain the last commanded value if the SmartServer cannot communicate to the device, or if it fails.

The default value is 0s.

cpControlCfg.minControlV (unsigned short)

It may be necessary to set the lowest 0-10V signal to a value that can be used to drive the controlled fixture. This value is the lower limit of the

control voltage driving the ballast/driver when the level is 0.5%. This value has a range of 0-250 with 0.04V resolution.

The default value is 0s.

cpControlCfg.maxControlV (unsigned short)

It may be necessary to set the highest 0-10V signal to a value that can be used to drive the controlled fixture. This value is the upper limit of the control voltage driving the ballast/driver when the level is 100%. This value has a range of 0-250, with 0.04V resolution. The default value is 250.

cpControlCfg.occupHoldTm (unsigned long)

The time that must elapse after seeing a SW_SET_OCCUPIED event before the SLC assumes the unoccupied scene level. The SLC will go to occupied levels when a SW_SET_OCCUPIED event occurs. It will stay in this state for at least the occupHoldTm duration. This value should be set to be two or three times longer than the heartbeat rate of the sensors.

The default value is 300s.

cpControlCfg.clrTime (unsigned long)

This field specifies the time in minutes after the dawn switch to OFF time that the latched alarm flags are cleared. The default value is 120 minutes.

cpControlCfg.lampType (olc_select_t)

Defines the lamp type.

Configuration Requirements/Restrictions

This CP has no modification restrictions (no_restrictions). It can be modified at any time.

SCPT Reference

SCPTcontrolCfg (382)

SLC Limits (Mandatory)

```
network input config sd_string("&1,p,0\x80,383")
SCPTlimitscpLimits;
```

This input configuration property sets the fault limits of the SLC.

cpLimits.powerFault (SNVT_lev_cont)

This is the percentage deviation allowed before power faults above or below what is expected based on the defined power profile for the luminaire under control by the SLC.

The default value is 15%

cpLimits.voltageFault (SNVT_lev_cont)

This is the percentage deviation allowed form the configured supplyVoltage before the SupplyVoltage alarms are triggered. See cpContro Cfg.supplyVoltage.

The default value is 15%.

cpLimits.lampVoltageFault (SNVT_lev_cont)

If supported, this is either the percentage deviation at 100% control level, or over the range in the case of a constant voltage controller for the voltage to the lamp.

The default value is 15%.

cpLimits.lampCurrentFault (SNVT_lev_cont)

If supported, this is either the percentage deviation at 100%, or over the range in the case of a constant current controller for the current to the lamp.

The default value is 15%.

cpLimits.pfLow (SNVT_pwr_fact)

Defines the power factor alarm point at which power factor levels measure below this value will generate an alarm condition. Power factor alarms are only tested when the controlled load is ON.

The default value is 0.65

cpLimits.rcvHb (SNVT_time_sec)

The rate at which the segment controller is expected to update nviLampValue or one of the other control input variables. Best practice is to set this value at 3 times shorter than cpControlCfg.rcvTmo

The Default value is 0s.

cpLimits.highTemp (SNVT_temp_p)

The temperature above which a high temperature alarm is triggered.

The defaults value is 65.0 C.

cpLimits.lampFailFault (SNVT_lev_cont)

The threshold of power drop measured when the lamp fails. In some technologies, induction lights for example, the power draw at bulb failure may be quite high. Controllers with hardware to detect lamp faults may not use this parameter.

The default value is 20%.

cpLimits.lampVoltage (SNVT_volt)

If supported, this is the expected drive voltage of the lamp at 100%, or over the control range in the case of a constant voltage driver.

cpLimits.lampCurrent (SNVT_amp_ac_mil)

If supported, this is the expected drive current for the lamp at 100%, or over the control range in the case of a constant current driver.

Configuration Requirements/Restrictions

This CP has no modification restrictions (no_restrictions). It can be modified at any time.

SCPT Reference

SCPTlimits (383)

Scene Table (Mandatory)

```
network input config sd_string("&1,p,0\x80,384")
SCPTsceneDef cpSceneTbl;
```

This CP defines a table (implemented as an array) of four to twelve scenes that can be recalled using nviLampValue. Each element is based on SCPTsceneDef which defines the scene number, level setting, and unoccupied scene. If it is desired to have a device not respond to the unoccupied condition, the unoccupied scene number should be set to 0.

Valid Range

none

Default Value

none

Configuration Requirements/Restrictions

This CP has no modification restrictions (no_restrictions). It can be modified at any time.

SCPT Reference

SCPTsceneDef (384)

SLC Location (Optional)

```
network input config sd_string("&1,p,0\x80,350")
SCPTgeoLocation cpLocation;
```

This configuration property sets the GPS location where the physical device is located. This is an optional cp because most procedures for installation do not modify this field.

Valid Range

SNVT_geo_loc

Default Value

none

Configuration Requirements/Restrictions

This CP has no modification restrictions (no_restrictions). It can be modified at any time.

SCPT Reference

SCPTgeoLocation (201)

Lighting Group Membership (Optional)

```
network input config sd_string("&2,1,0\x80,361")
SCPTlightGroupMembership cpGroupMember;
```

This configuration property is used to assign the SLC to one or more lighting groups to define the controller's response to updates to nviLampSw2 when the state field is set to SW_SET_GROUP_STATE_LEVEL (23) (first defined in 13.20)

Valid Range

SCPTlightingGroupMembership

Default Value

none

Configuration Requirements/Restrictions

This CP has no modification restrictions (no_restrictions). It can be modified at any time.

SCPT Reference

SCPTlightingGroupMembership (361)

Backup Schedule (Optional)

```
network input config sd_string("&1,p,0\x80,344")
SCPTbkupSchedule cpBkUpSchedule;
```

This configuration property defines a default on/off schedule for cases when the lamp controller detects that communication with the segment controller is lost.

If the START and the END time are the same, then there is no backup schedule defined.

Valid Range

SCPTbkupSchedule

Default Value

none

Configuration Requirements/Restrictions

This CP has no modification restrictions (no_restrictions). It can be modified at any time.

SCPT Reference

SCPTbkupSchedule (344)

SLC Power Profile (Optional)

```
network input config sd_string("&1,p,0\x80,381")
SCPTpowerProfile cpPowerProfile;
```

Defines the typical power measured at 5 commanded nviLampValues (0.5%, 25%, 50%, 75%, 100%) for the actual driver/lamp combination. This table is used to determine expected power at various light levels for high and low power alarms. SNVT_power is the type used for the elements in this table. The values must be determined after the minPWM and maxPWM fields defined in the following section are set, and should be done at the intended operating voltage.

Valid Range

SNVT_power

Default Value

none

Configuration Requirements/Restrictions

This CP has no modification restrictions (no_restrictions). It can be modified at any time.

SCPT Reference

SCPTpowerProfile (381)

Key for Unresolved References

p is this Object's index relative to the node sd_string declaration, when implemented.

Data Transfer

None specified.

Power-up State

None specified.

Boundary and Error Conditions

None specified.

Additional Considerations

The original Outdoor Luminaire Controller profile lacks support of two important features that were not possible to realize using available using master/slave control and power line repeating.

1. Broadcast level control.
2. Peer-to-peer driven Traffic/Occupancy events.

These features require changing the primary control variable to use SNVT_switch_2. This section provides a detailed explanation of how the SLC uses this type to leverage these important control scenarios.

Scheduling is now accomplished using device specific scene levels for different periods of operation. The SLC should provide support for action based on the SNVT_switch_2 `state` values shown in table 1 below.

Table 1 - SNVT_swith_2 states recognized by the SLC.

State Enum	Value	Notes
SW_NUL	-1	Invalid value. The device may initialize the state field to this value, and assume the default level on reset.
SW_SET_LEVEL	5	Set control level.
SW_RECALL_SCENE	8	Recall a specified Scene level.
SW_SET_OCCUPIED	11	When an SLC is scheduled to an ON level or for operation, this state restores the current Scene Level.
SW_SET_UNOCCUPIED	12	When the occupancy hold timer has expired, the SLC will play the Unoccupied level if the SLC is scheduled to an operating scene.

The primary input variable, nviLampValue, is used by a single scheduler to set light levels, and to respond to occupancy events from one or more Traffic/Occupancy Sensor (TOS) devices in the system. It is assumed there is one and only one device setting the schedule, and potentially many devices sending traffic/occupancy events. A `state` value of `SW_RECALL_SCENE` is used to schedule control levels in a streetlight segment because it allows a single update using a group addressed broadcast to control all SLCs managed on a segment while supporting different control levels defined by the scene table programmed in each SLC. Scenes numbers are limited to the range of 1-255. The `scene_number` 255 has special meaning in SLC applications. It is used to schedule the SLC to the daytime OFF condition. When the SLC is scheduled to daytime OFF, no action is taken on `SW_SET_OCCUPIED` of `SW_SET_UNOCCUPIED` states coming from TOS devices. It is important to distinguish daytime OFF from scheduled OFF condition. Certain applications may require the SLC to switch the load OFF when the unoccupied condition is determined.

The SLC includes a scene table array that supports the scene definitions as described in this structure.

```
typedef struct {
      unsigned short scene_number;
      SNVT_lev_cont  setting
      unsigned short unoccupied_scene;
}SNVT_scene_Def;
```

The scene table is sized to have between 4 and 12 entries based on developer preference and available memory..

The scene_number field can take a value in the range of1-254. A value of 0 defines a table entry as not used. A value of 0 in the unoccupied_scene level configures the SLC to not respond to unoccupancy events. Scene_number values may define different type of SLC response. This is best understood by

describing use cases in the following sections. As mentioned previously, a value of scene_number of 255 is reserved to switch OFF the load during the day and signal the SLC to not respond to occupancy events.

Intersection Locations

For safety reasons, the light levels for SLCs located in intersections are not allowed to dim to reduce energy levels. In this case, the SLCs are schedule ON and OFF against the sunrise/sunset calculations of the SmartServer. Levels are maintained regardless of traffic conditions.

Table 2- Scene table for an SLC used in intersection locations.

Table entry	scene_number	Setting	unoccupied_scene
cpSceneTbl[0]	1	200	0
cpSceneTbl[1]	2	200	0
cpSceneTbl[2]	3	200	0
cpSceneTbl[4]	0	0	0

Based on this scene table configuration, the segment controller would define two presets as shown in table 3.

Table 3 - SmartServer schedule presets.

Preset	Value	Notes
DUSK_ON	SW_RECALL_SCENE, 0, 1	Scene definitions as the node could result in different control levels based on application need. Note, the feedback SNVT_switch_2 variable will report the value.
DAWN_OFF	SW_RECALL_SCENE, 0, 255	Scene 255 has a reserved meaning for daytime OFF. Because it is reserved, a scene table entry is not required.
ON_MED	SW_RECALL_SCENE,0 , 2	SLCs that do not have a scene table entry defining scene 2 take no action.
ON_LOW	SW_RECALL_SCENE, 0, 3	

With the intersection use case, the SLC only switches between 100% and OFF. Any SW_SET_OCCUPIED and SW_SET_UNOCCUPIED updates for TOS devices will not change the output of the intersection lights. Scheduled energy reducing updates will also have no effect because the scene table does not define scenes 2 and 3 used by the SmartServer for energy saving dimming profiles.

It is important for the SLC that scene table is padded out to define the same output level for scene_number(s) 1, 2, and 3. If the SLCs are commanded using a single point on the segment controller that is broadcast to all SLC devices using a heartbeat, a reset of an individual LC will get the level on track within the heartbeat interval. If the SLC has only a definition for scene_number 1, and it experiences a reset while the current scene being played by the segment controller is 2, the node would not go to the intended level on the heartbeat interval. If a device receives an update recalling a scene that is not in the scene table, no control action is taken.

Basic Dimming Profile

The SLC will be scheduled by the SmartServer to reduce light levels at various times when traffic is typically light. Lights do not respond to SW_SET_OCCUPIED/SW_SET_UNOCCUPIED commands from TOS devices.

Table 4 - Basic scheduled dimming scene table.

Table entry	scene_number	setting	unoccupied_scene
cpSceneTbl[0]	1	200	0
cpSceneTbl[1]	2	150	0
cpSceneTbl[2]	3	100	0
cpSceneTbl[3]	0	0	0

The scene definitions defined in table 4 are used to schedule a dimming profile. It is important to understand that SLCs using SNVT_switch_2 may be controlled using a one to many binding from a SNVT_switch_2 defined dynamic point on the segment controller. This creates rapid switching of many targets nearly together rather than sequentially as was the case prior to SmartServer release 2.2 when there was no support for peer binding.

Dimming Response with TOS device Signals

In this use case, the SLC will respond to TOS signals. Two UNOCUPIED levels are used depending on the schedule.

Table 5 - TOS responsive SLC.

Table entry	scene_number	setting	unoccupied_scene
cpSceneTbl[0]	1	200	10
cpSceneTbl[1]	2	150	11
cpSceneTbl[2]	10	100	0

cpSceneTbl[3]	11	60	0

In this application, the SmartServer will schedule to light ON with scene 1. At a later time when traffic is typically lighter, the SmartServer will schedule scene 2. Scenes 1 and 2 reference different UNOCCUPIED scene numbers to limit the visual steps between the OCCUPIED and UNOCCUPIED levels. This configuration is used to show the flexibility of the approach. A single UNOCCUPIED scene could be used.

Device Design Recommendations

Power line networks must consider the limited bandwidth of the power line. This is straight forward to control when the SmartServer controls and monitors SLCs using a master/slave control paradigm. The goal is to seek to limit the traffic be less than 4 packets/sec. There are also specific device resource requirement.

The following design considerations are necessary.

1. Peer-to-peer updates from TOS devices should not be offered faster than every 300s, except in the case of the active action (SW_SET_OCCUPIED).
2. Complex repeating topologies with large numbers of sensors (more than 6), may require additional reductions in to 300s rate described in the previous rule.
3. If the sensor coverage has a high degree of overlap, using UNACKD service may provide faster response.
4. Using ACKd service for peer-to-peer connections only applies if the connection is one-to-one. Group a connections use only UNACKD_RPT, or UNACKD service.
5. SLC devices must include 3 or more address table entries.
6. SLC devices must include an alias table entry for each TOS device it is connected to in excess of one. If there are 5 TOS devices providing occupancy signal, 4 alias table entries are required.
7. Network variable updates must be idempotent. This means an update if received multiple times in close succession will not impact the result. This is the case with SW_SET_OCCUPIED or SW_SET_UNOCCUPIED updates. An example of an update that is non-idempotent is an update that requests a toggle of state, or percentage change in level.
8. The source address of message may not reflect the actual source of the update. The rebroadcast mechanism causes this information to be lost.
9. The SLC transition from OCCUPIED to UNOCCUPIED will be time based, and should allow for lost updates. For example, if the TOS devices generate 300s heartbeats, the SLC should allow 900s to pass without observing a SW_SET_OCCUPIED command before taking the action to reduce light levels.
10. SLCs controlled scheduled by a segment controller SNVT_switch_2 output broadcast to the group must have scene definitions for the scenes recalled by this data point. This is required to allow an SLC to reset and end up with the correct level once the segment controller heartbeats the value.
11. Behaviors for transitions between scenes require considering several factors.

a. From daylight OFF to ON, the SLC should assume OCCUPIED levels if the SLC does not detect a valid TOS source.
b. If the SLC is scheduled to transition from a scene which includes an UNOCCUPIED level, a transition of a new scene will cancel the UNOCCUPIED state if the new scene does not have an UNOCCUPED level defined. If an UNOCCUPIED scene is defined for the new scene, the level of the new scene's unoccupied scene will be used.
c. If the SLC is scheduled to transition to transition from a scene which does NOT have an UNOCCUPIED scene assigned to one that includes an UNOCCUPIED scene, the state at the transition is assumed to be OCCUPIED, and the timer to UNOCCUPIED is started.
d. The transition to scene 255 (daytime OFF), is immediate, OCCUPANCY state timers processing is stopped, and response to SW_SET_OCCUPIED and SW_SET_UNOCCUPIED is ignored.

The assumptions described in this section assume a profile for an Occupancy Sensor profile which using SNVT_switch_2 to issue SW_SET_OCCUPIED and SW_SET_UNOCCUPIED state values to the SLCs. The specific use case described for demand driven applications describe above allows for simple to manage peer-to-peer relationships. The segment controller binds one or more defined SNVT_switch_2 outputs to drive as many independent scene based schedules as need in the system. Bindings from the TOS sensor, to the SLCs are also simple to realize. The use case described in this document is quite narrowly defined to specifically enable responsive sensor driven lighting co