



LONMARK[®]

Functional Profile:

Real Time Based

Scheduler

3301-10 © 1997, LONMARK Interoperability Association

Echelon, LON, LONWORKS, LONMARK, and the LONMARK logo are trademarks of Echelon Corporation registered in the United States and other countries.

Overview

The scheduler object sends events over the network that are based on real time. These events are used by controllers to perform actions based on the actual time of the day (e.g. at 6 pm switch OFF the lights). Optionally this object will also provide time related building status information, like normal working hours, overtime, cleaning period, etc..

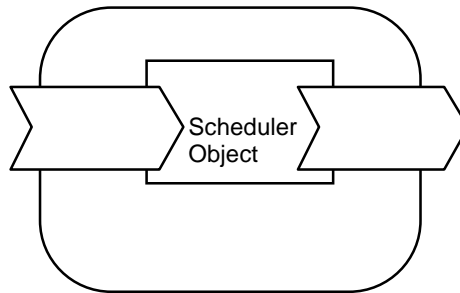


Figure 1.1 Scheduler Functional Profile

Example Usage

The scheduler object provides a controller function. In principle it can be used as a single object on a device, or in conjunction with other objects such as the time keeper object. The scheduler object receives its time input from a Real Time Keeper object. The mandatory output of the scheduler object is `nvoScene` which can be used by actuators and other controllers to initiate real time related events. Optionally the output `nvoSetting` can be included to enable actuators and controllers to take actions in relation to real time related operational modes such as working hours and cleaning hours.

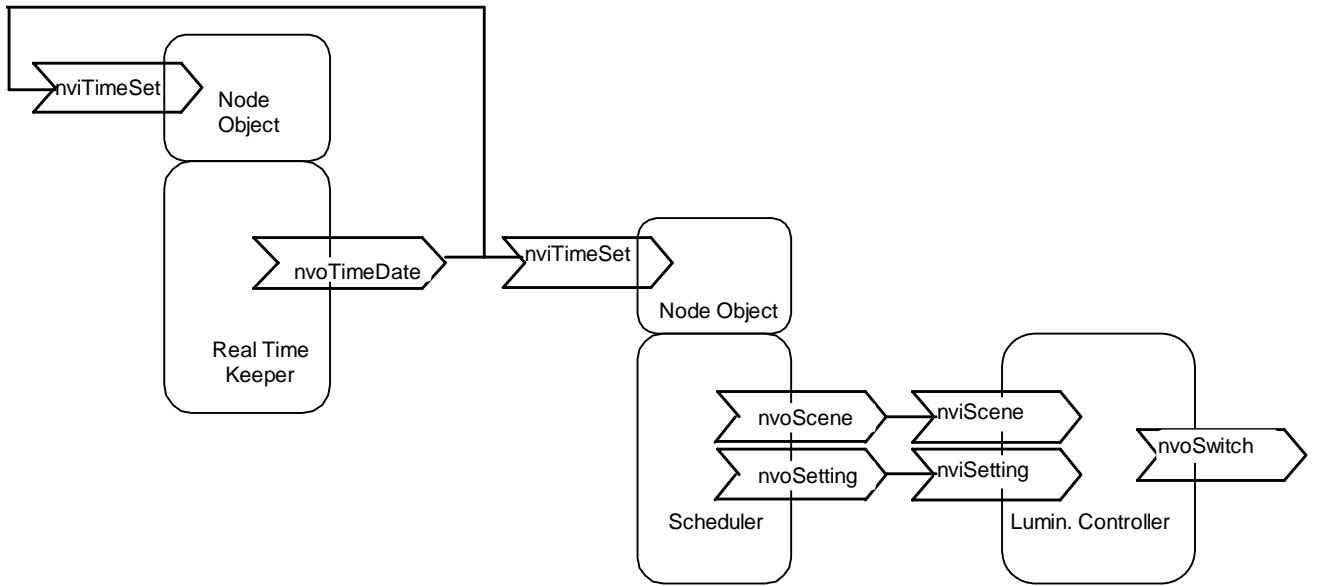


Figure 1.2 Example Usage

Object Details

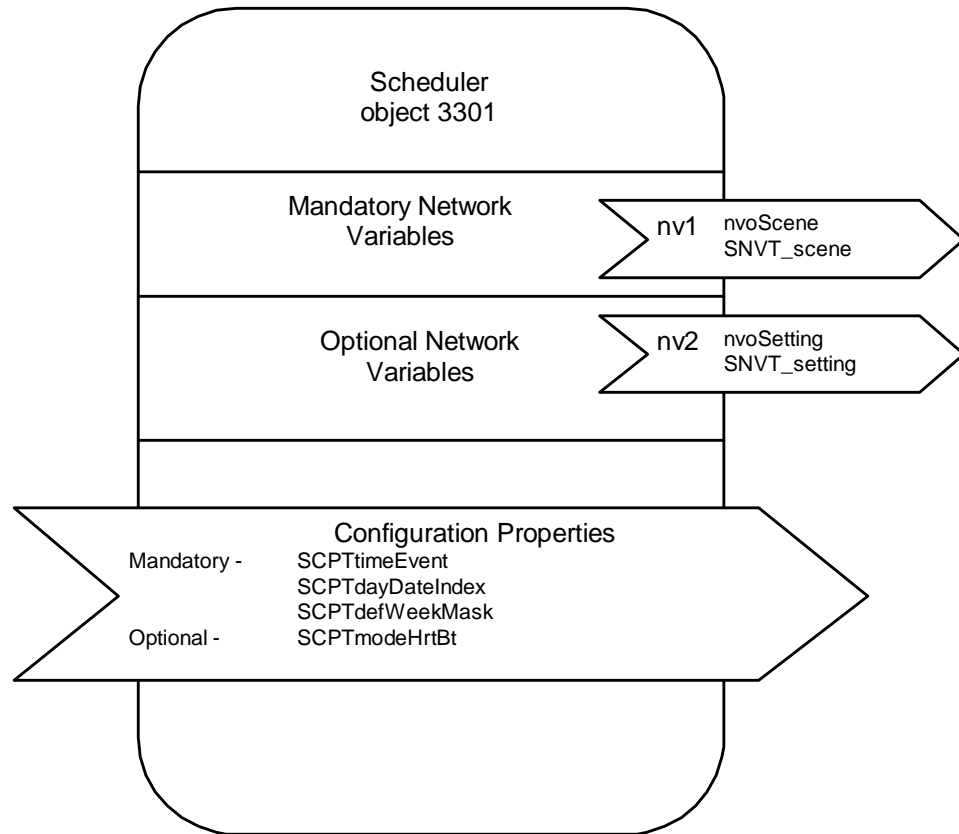


Figure 1.3 Object Details

Table 1 SNVT Details

NV # (M/O)*	Name	In/Out	SNVT Type (SNVT Index)	Class	Description
1 (M)	nvoScene	Out	SNVT_scene (115)	nv	A time related event
2 (O)	nvoSetting	Out	SNVT_setting (117) ¹	nv	A time related building operation mode

* M = mandatory, O = optional

Table 2 SCPT Details

SCPT index (M/O)*	Name	Description
102 (M)	SCPTdefWeekMask	Week Mask
103 (M)	SCPTdayDateIndex	Days - dates pointer [array]
104 (M)	SCPTtimeEvent	Time event-mode tuples [array]
105 (O)	SCPTmodeHrtBt	Heart beat for mode output

* M = mandatory, O = optional

Mandatory Network Variables

There is only one mandatory network variable `nvoTimeEvent` of the type `SNVT_scene`.

Time Event Output (new SNVT)

network output `SNVT_scene nvoScene;`

This output network variable provides a time and date related event for the light controller. It is up to the light controller to take appropriate actions (if any).

Valid Range

Range of `SNVT_scene`.

When Transmitted

Depending on the day-type and time of day, once when the defined time of day is reached.

Update Rate

Each time a new event is found in the schedule. There is no heart-beat on the scene event.

Default Service Type

Depending the number of controllers connected to the output network variable,

if less than 64 -> acknowledged,

if more or equal than 64 -> unacknowledged - repeated.

Optional Network Variables

There is one optional output network variable `nvoSetting` of the type `SNVT_setting`.

Mode Output (new SNVT)

network output SNVT_setting nvoSetting;

This output network variable provides real time related operational mode information for the lighting controller. This operational mode information makes e.g. following application possible: switch ON lighting during Cleaning hours, but not during Patrol hours.

Valid Range

Range of the SNVT_setting;.

When Transmitted

At the start of a mode period as defined in the data base.

Update Rate

Optional this output can be provided with an heart-beat, depending the configuration parameters.

Default Service Type

Depending the number of controllers connected to the output network variable,

if less than 64 -> acknowledged,

if more than or equal than 64 -> unacknowledged - repeated.

Mandatory Configuration Properties

To enable proper functioning the scheduler object will contain a data base. The data base contains two SCPT arrays and one SCPT. Size of the arrays are defined in the Configuration Parameter Template file.

The first array is an array with SCPTtimeEvent. The array will hold "terminated" lists of time_event_mode definitions (event = scene, mode = operational; mode). Each list will hold the events and the modes to be executed on a defined type of day.

The second array is an array with SCPTdayDateIndex. It consist out of date-dates definitions with an index to the first time_event_mode definitions array. The object shall scan this list for an occurrence of the actual day (date) to find the appropriate schedule for that day. When date(s) are multiple defined the first occurrence in the list has priority.

If no date is found in the second array the SCPTdefWeekMask defines a default list for that day of the week.

It is up to the configuration tool to optimize the use of the array's.

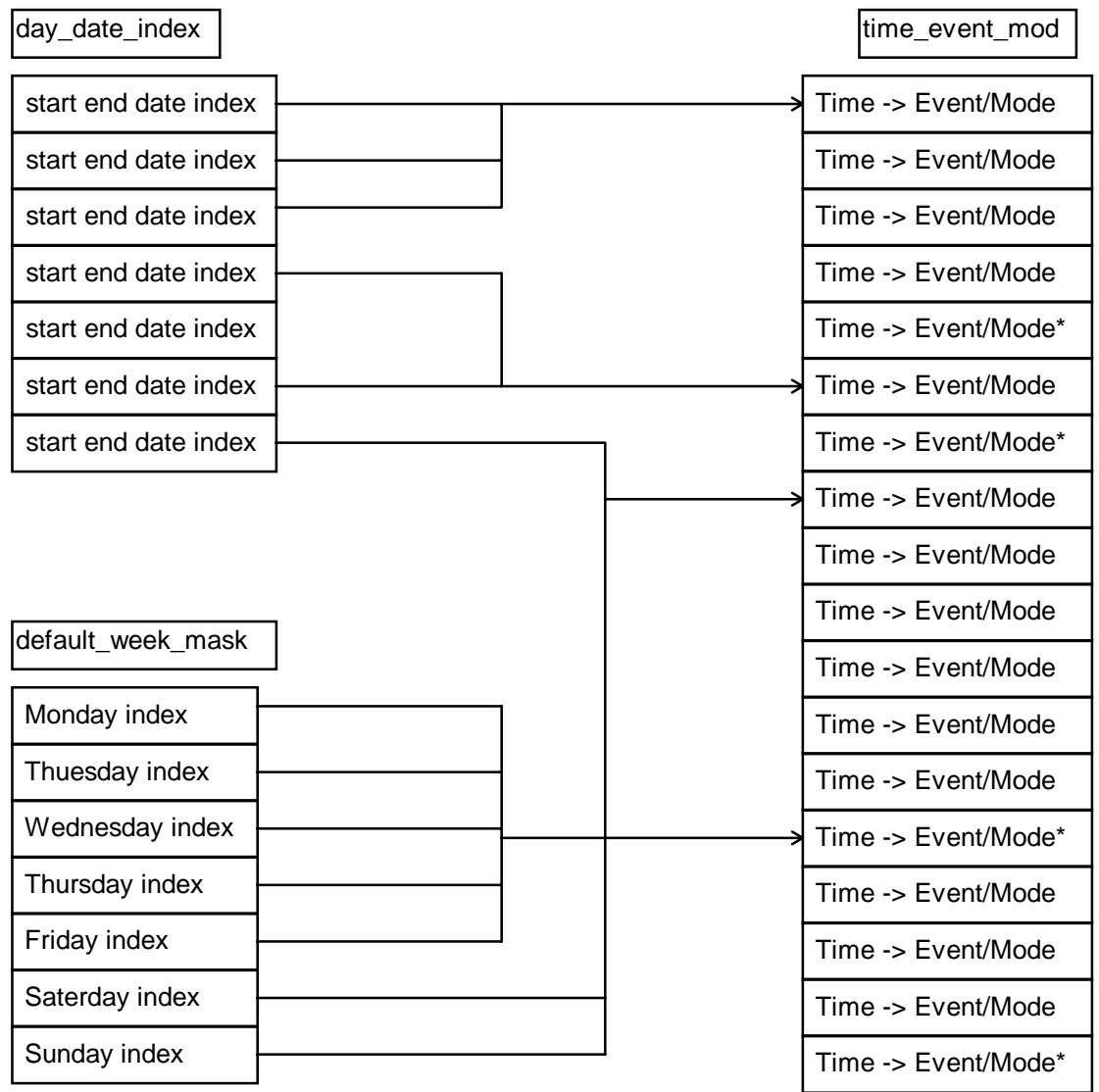


Figure 1.4 Data Base Structure

NOTE: For the elements Time -> Event/Mode*, the * indicates the end of list indicator.

Time event/mode array element

This input configuration property provides event or mode definition to be transmitted on the network if the time in the tuple is reached.

Valid Range

The valid range of the SCPT

Default Value

List terminator.

SCPT Reference

SCPTtimeEvent #104

SCPT elements:

record type (0 = end of list, 1 = scene, 2 = mode)
hour (0 ... 23)
minute (0 ... 59)
event_mode (0 ...255)

Day-Date Index array element

This input configuration property provides one or two dates together with a start pointer to the Time-Event array.

Valid Range

The valid range of the SCPT

Default Value

Empty (date 0/0)

SCPT Reference

SCPTdayDateIndex #103

SCPT elements:

day_1 (0 ... 31)
month_1 (0 ...12)
day_2 (0 ... 31)
month_2 (0 ... 12)
event_mode_index (0 ... 64K)

Default Week Mask

This input configuration property provides a day type definition for every day of the week

Valid Range

7 elements with a day type for each element

Default Value

not applicable

SCPT Reference

SCPTdefWeekMask #102

array of 7 (seven) longs, long value is the start index in the time_event_mode array

long [0] = index for Monday (0 ... 64K)
long [1] = index for Tuesday (0 ... 64K)
long [2] = index for Wednesday (0 ... 64K)

long [3] = index for Thursday (0 ... 64K)
long [4] = index for Friday (0 ... 64K)
long [5] = index for Saturday (0 ... 64K)
long [6] = index for Sunday (0 ... 64K)

Optional Configuration Properties

Mode heart-beat

This input configuration property ble provides mode definitions to be re-transmitted if the time in the tuple is reached.

Valid Range

valid range of SNVT_time_sec

Default Value

No heart beat.

SCPT Reference

SCPTmodeHrtBt #105.

Data Transfer

There is no other than configuration data to be transferred.

Power-up State

No immediate network action on Power Up state. Optional `nvoTimeMode` shall be updated on next occurrence of heart beat.

Boundary and Error Conditions

None.

Additional Considerations

During update of the configuration data base the object shall be set Off Line.

When multiple objects are implemented on the same node it should be possible to share the `event_mode` array between the objects.