LITHONIA LIGHTING

A Guide to Integrating Synergy Lighting Control Into Building Automation and Control Systems Using ANSI/ASHRAE Standard 135-1995, BACnet.

SYNERGY™

Revised 2/03

Table of Contents

Abstract

This guide is intended for use by systems integrators or design consultants in specifying or installing BACnet control systems that will include lighting control using the Synergy lighting control system by Lithonia Lighting. It explains the relationship between Synergy lighting control functions and the BACnet device object structure. This document should be considered as "typical" in nature with the focus being on issues relating to the interconnection and interoperation of Synergy with the building automation system using BACnet. It is intended to complement other more general guidelines for direct digital control systems for building automation applications. Aspects of the Synergy system that may be pertinent to lighting control functionality but not relating to integration are not necessarily discussed in this document.

Product Description

A BACnet Synergy system consists of one or more lighting control cabinets that are usually interconnected via a twisted pair wire buss. The system may or may not include a personal computer with Synergy application software. While it is possible to configure the system for network operation through the user interface panels (UIP) on the system controllers, users may find it advantageous to use the available Windows application software to configure multiple cabinet systems.

There are three basic component types that make up the Synergy system: enclosures (catalog sheet *SY200*), power modules (catalog sheets *SY400, SY410 & SY420*), and the system controller (catalog sheet *SY320*). Synergy SYE series enclosures are shipped with an integral power supply and internal ACCESS.bus™ interface and cable for connection of the power modules. Each cabinet that is to function as a node on the control network will have a MLX Series system controller installed (see *Synergy Operation and Programming Manual*) with the exception of a secondary cabinet used to expand the input/output point count for a single system controller.

The power modules are I/O devices each containing a series of low voltage (24 VDC) inputs and line voltage outputs. SYPM Power Modules installed in the Synergy cabinet will be configured for either switching only (relays) or dimming with on/off switching (dimmers with series relays). Each power module will contain some combination of switch input terminals and analog inputs with a source of 24 VDC accessory power. The most common use for the switch input terminals if for the connection of low voltage switches to manually control lighting or override the program schedules. The most common use for analog inputs will be for connection of a photocell to read and report the exterior ambient light level.

The standard physical communication layer for Synergy is RS485, ARCNET at 156 k baud. The ARCNET port is located on the side of the MLX system controller in the form of a removable terminal block assembly. All rules for implementing an ARCNET communication network must be followed. Termination jumpers are provided within the controller for proper end-of-line terminations. If required, Synergy controllers can be connected to an ETHERNET backbone through the use of appropriate routers.

Input/Output Architecture

General

Functionally, Synergy is an event driven system. The logic engine in the Synergy controller recognizes input events. It evaluates the events in relation to a native hierarchy based on BACnet priority levels. Then generates an output event that usually affects the state of one or more relays or dimmers. Input events are generated from a variety of sources. These may include the pressing of a switch by a user, an internal Synergy time schedule event, or a digital command received via the network. Since the BACnet standard does not specifically address lighting, we have assigned priority levels within our logic structure as appropriate to our needs for lighting control. *It is important to carefully consider the use of Priority_Level when writing present values to a Synergy point* (see Priority). It should be noted that internally, Synergy uses the Analog Output Object for all outputs regardless of their physically being a relay or a dimmer. In order to be user friendly to those expecting a relay to be a binary device, relays are also recognized as Binary Output Objects. Therefore, a lighting control panel will exhibit parallel point lists where BO and AO objects of the same instance number represent the same physical output (relay or dimmer).

Relays and Dimmers (Binary Output Object Type/Analog Output Object Type)

The relays and dimmers are the actual devices that will ultimately control the lighting circuits. Relay power modules each contain eight output points while dimmer modules contain six. The Instance_Numbers of these points are organized in a "hotel" numbering scheme based on the ACCESS.bus address of the module. As an example, a module with the buss address of "01" (see *installation instructions* for module rotary address switch setting), will contain output points 0101 – 0108. Module address "08" will have points 0801 – 0808 etc. Synergy lighting control documentation refers to these output points as relays or dimmers. In actuality*, the individual Synergy output points are Binary Output Objects and Analog Output Objects. You may individually set or read the Present_Value of these points in the normal manner.*

Switch Inputs (Binary Input Object Type)

Switch inputs on a Synergy power module are a series of spade terminals that are provided for the connection of low voltage switches. These switches provide a means for building occupants to signal the system to turn lighting on or off. Relay modules each provide eight switch inputs, while dimmer modules each provide two. The Instance_Numbers of these inputs are organized exactly like the outputs using the hotel numbering scheme (see Relays and Dimmers). *As far as BACnet is concerned, Synergy switch inputs are Binary Input Objects which are available to external systems for read/write of Present_Value.* When a switch is pressed, it causes the controller to write a Present_Value of 100.0 for ON or NULL for OFF to a Group of outputs that have been specified in the present controller configuration (see Groups).

Analog Inputs (Analog Output Object Type)

Analog inputs on a Synergy module are a series of spade terminals that are provided for the connection of 0 – 10 VDC analog devices such as the LSA APS photocell. Each relay module has two analog inputs, while a dimmer module has three. The Instance_Numbers of these inputs are organized exactly like the outputs using the hotel numbering scheme (see Relays and Dimmers). *As far as BACnet is concerned, Synergy analog inputs are Analog Input Objects.* Reading the Presetnt_Value will return a value in the range of 0.0 - 100.0. With the LSA APS OL photocell installed, this will equate to 0 – 100 foot-candles of light.

Group (Analog Value Object Type)

The Group is a convention used in the Synergy controller configuration to organize lighting loads into logical zones for more convenient operation and programming. Relays and/or dimmers put into a Group, may be acted upon collectively with a single action. As an example, the corridor lighting in a building might be connected to several relays. If these relays are combined into a Group in the Synergy controller configuration, it will require the reading/writing only a single point to control and monitor the corridor lighting. *To BACnet, a Synergy Group is represented as an Analog Value Object. In the above example, as an alternative, you could read/write the Binary Output points representing the corridor relays individually.* An advantage of working with individual points is that it requires no field configuration of the Synergy controllers. The advantage of using the Group method (Analog Value) is response speed and reduced number of network messages. When writing to a Group, the Present_Value and Priority are passed on to all members of the group.

If the state of member relays or dimmers are changed by independent action, the Analog_Value point representing the Group will return a Present_Value equal to the average of the states of the member relays. An AV representing a group containing four relays with three on and one off would return a Present Value of 75.0.

Priority (Priority Property)

The BACnet standard does not specifically address the aspect of lighting control. It was therefore necessary for us to make some determinations on our own when implementing BACnet in the design of Synergy. We use the following local priorities in Synergy:

```
#define PRIO_PRIORITY_ON        3    /* Synergy Unique */
#define PRIO_PRIORITY_OFF       4    /* Synergy Unique */
#define PRIO_MANUAL_OPERATION   8    /* only used with flash to find */
#define PRIO_NORMAL_OPERATION   10   /* Synergy Unique */
#define PRIO_PRIORITY_LOW       13   /* Synergy Unique */
```

Priority ON will relinquish on OFF (0.0). Priority OFF will relinquish on ON (100.0). Normal and priority low relinquish on OFF. Flash-to-find relinquishes on deactivate. Normal relinquishes on OFF (0.0). If there is need to turn a load OFF without relinquishing the priority, do not write a Present_Value of 0.0. Instead, write a value of 1.0 (or any value less than 50.0).

It is necessary for Synergy to relinquish priorities under the above conditions in order to achieve the desired functionality as a lighting control system. Synergy will also respond to *relinquish* from a workstation or from the integral command line interpreter. Performing a manual override from the user interface panel on the Synergy controller will also relinquish all but Priority 1 and 2.

It is important to note that when writing to points in a Synergy panel, the command priority that is used can have a detrimental effect on the independent operation of the lighting control system. It is strongly suggested that writes intending to turn lighting ON be made at command priority 10 and writes intending to turn lighting OFF write a NULL unless the intention is to forcibly override the normal functioning of the lighting controls. If the internal lighting control functionality of Synergy is not being used (Synergy is a slave to the host system), you are free to write to the points as you wish.

As an example, a lighting load is connected to Synergy relay #104 which in turn, is controlled by a low voltage wall switch using internal Synergy functionality. Synergy uses priority level 10 for normal switch operation. The BAS then turns the lights on by writing a Present Value of 1.0 at command priority 9 to BO #104. The Synergy switch will now not be able to turn the lights OFF since the priority array contains an ON at level 9.

Other Input/Output Options

Telephone Interface

The Synergy PHONE option allows the user to enter a zone code and level via touch-tone telephone to control lighting. This is accomplished through the writing of the Present_Value Property to an Analog Value Object (Group). It is also possible to use the PHONE option to control other BACnet devices (see **Writing Values to Other BACnet Devices).**

Writing Values to Other BACnet Devices

The Synergy controller is factory configured to screen on Vendor_ID number 42 (Lithonia Lighting). It is possible to field enable Synergy to write Present_Value to Analog Output Objects, Binary Value Objects and Analog Value Objects in other manufacturer's devices by first editing the *config.ini* file in the Synergy controller (see RS232 Interface). As an example, the following entry would enable Synergy to write to *Alerton* and *Automated Logic* devices.

```
[bacnet]
vendor=18
vendor=24
```

A group is created in the set up of the local Synergy controller that contains the remote point to be controlled. The remote point is represented as a group (AV), dimmer (AO), or partition (BV) in a remote Synergy controller. The remote Synergy controller is actually the third party BACnet device and the point to be controlled is the appropriate instance of AV, AO or BV.

DMX512 Control

The Synergy controller can act as a gateway between a theatrical control device and a BACnet network. A Synergy controller equipped with the DMX interface option is capable of mapping up to 512 channels of DMX (one universe) to a series of Analog Value objects. Synergy monitors the DMX stream and creates an event (writes a Present_Value to a point) whenever a change in level occurs on a DMX channel.

RS232 Interface

The Synergy controller is equipped with a command line interpreter that enables a PC connected to the RS232 port on the controller to communicate with the BACnet network using terminal emulation. The connection requires a null modem serial cable connection between the PC and the Synergy controller. Windows HyperTerminal settings are as follows:

```
57,600 baud (bps)
8 data bits
none parity
1 stop bit
ANSI terminal emulation
hardware flow control
```

Pressing <Enter> on the PC keyboard will return a line similar to the following indicating that communication has been established.

```
SYNERGY Revision v#.## built on day-mon-year
```

```
>
```

Typing `bacnet <enter>` at the `>` prompt will return a list of commands as follows:

```
>bacnet
 upgrade [synergy.rtb] <id>[,id,id,id] - upgrade many controllers
 dir - list of file handles and names
 timesync - synchronize the network time with my time
 timemaster [minutes] - time master sync occurs every x minutes
 recipient [id] - adds a time sync recipient to list
 id [id] - read[write] the BACnet device object id
 arcid [id] - read[write] the ARCNET id
 mstp - display MS/TP state maching information
 dcc <id> <0/1> [duration] [password] - device communication control
 whois [id] - Who-Is command
 whohas <[name|type] [instance]> - who has object
 vendor [id] - show vendor id list [temp add id]
 reinit <id> <password> [warm]
 reboot <id> - Re-Init Device id
 bootdisk <id> - change boot application for id
 objects <id> - list the device objects for id
 dumpall - display devices who responded to Who-Is
 dump - display devices in our address cache
 q - display number of messages waiting to be sent
 status - show status of ARCNET driver
 time <hh:mm:ss mm/dd/yyyy> - set network time
 [commands]
 <id> relay <#> [val] [prio] - write/read relay value at id
 <id> dimmer <#> [val] [prio] - write/read dimmer value at id
 <id> partition <#> [val] [prio] - write/read partition value at id
 <id> group <#> [val] [prio] - write/read value to group at id
 <id> switch <#> [val] - read[write] a value from switch at id
 <id> analog <#> [val] - read[write] a value from analog at id
 <id> fileread <#|name> <name> - read file# from id, save to name
 <id> filewrite <#|name> <name> - write file name to id as file#
 <id> read <#> <type> <property> [index] - read property
 <id> priority <instance> [AO|BO|AV|BV] - read priority array
```

To write a value of 75.0 at priority 10 to AV instance 12 in device ID #4217 type:

```
bacnet 4217 group 12 75 10 <enter>
```

To return the value of AO instance 1 in device ID #18211 type:

```
bacnet 18211 relay 1 <enter>
```

Note that the command structure uses the proprietary Synergy terms group and relay rather than Analog Value and Analog Output.

LIGHTING PANEL SCHEDULE

Project: _____Page ____ of _____
Panel Name _____Panel Location _____
Device ID: #42_____ ARCNET # _____ Network _____

BI (Switch Input) Factory Default		**AV** (Group) User Assigned		**BO/AO** (Relay or Dimmer) Factory Default	
Instance #	**Name**	**Instance #**	**Name**	**Instance #**	**Name**
101				101	
102				102	
103				103	
104				104	
105				105	
106				106	
107				107	
108				108	
201				201	
202				202	
203				203	
204				204	
205				205	
206				206	
207				207	
208				208	
301				301	
302				302	
303				303	
304				304	
305				305	
306				306	
307				307	
308				308	
401				401	
402				402	
403				403	
404				404	
405				405	
406				406	
407				407	
408				408	
501				501	
502				502	
503				503	
504				504	
505				505	
506				506	
507				507	
508				508	
601				601	
602				602	
603				603	
604				604	
605				605	
606				606	
607				607	
608				608	