LITHONIA LIGHTING
ELECTRONIC SYSTEMS GROUP

Synergy Script Logic

Design Specification

Lithonia Control Systems
Lithonia Lighting
Acuity Brands Lighting

Issue: 2.65

Lithonia Lighting
One Lithonia Way
Conyers, Georgia 30012

Printed on November 30, 2006.

Document Information

Author(s): Steve Karg

Filename: script-logic.doc

Change History

Issue	Date	Description
1.0	16 Sep 1999	Initial draft –Steve Karg
1.1	03 Nov 1999	Added more examples. –Steve Karg
1.2	04 Apr 2000	Added a new expression and a new command. –Steve Karg
1.3	28 Jun 2000	Added POWERUP expression and example. - Steve Karg
1.4	18 Jul 2000	Removed GROUP from expressions, and PRIORITY from commands because they weren't working in the code. - Steve Karg
1.5	28 Feb 2001	Added CLEAR to SCHEDULE action. - Steve Karg
1.6	11 March 2002	Added ALARM - Steve Karg
1.7	01 July 2005	Updated for version 2.63 of Synergy.
1.8	20 October 2005	Added remote device commands for version 2.63 of Synergy. - Steve Karg.
1.9	30-November-2006	Added enable/disable warn, and included additional examples. - Steve Karg

References

None

Table of Contents

INTRODUCTION

The Synergy product will be capable of interpreting the macro syntax (known as Script Logic) described below and performing logical operations defined by the user. The script logic will be event driven, such that after an event happens, logic is probed to see if expressions are valid.

The basic construct of the script logic file, LOGIC.TXT, is as follows:

```
LOGIC(1.0)

// This line is a comment
IF expression THEN
  command
ELSE
  command
  command
ENDIF

IF expression OR expression THEN
  command
  command
ELSE
  command
ENDIF

IF expression AND expression THEN
  command
  command
ELSE
  command
  command
ENDIF

IF expression THEN
  IF expression THEN
    command
  ENDIF
  command
ELSE
  command
  command
ENDIF

IF (expression OR expression) AND expression THEN
  command
ENDIF
```

EXPRESSIONS

The following is a list of expressions. These belong inside the IF-THEN statement. The operand can also be another object and id, in which case the value of the second object is used for comparison.

OBJECT	ID	TEST	OPERAND
SWITCH ANALOG BUTTON PHONE DMX	####	=,>,>=,<,<=,<>	0-100 ON OFF
RELAY DIMMER LEGACY-DIMMER DALI DALI-GROUP PARTITION	####	=,>,>=,<,<=,<>	0-100 ON OFF
~~GROUP~~	~~####~~	~~=,>,>=,<,<=,<>~~	~~0-100~~ ~~ON~~ ~~OFF~~
RELAY	#### STARTS	=,>,>=,<,<=,<>	###### - up to 6 digits representing number of starts RELAY #### STARTS [2.62]
RELAY	#### HOURS	=,>,>=,<,<=,<>	###### - up to 6 digits representing number of hours RELAY #### HOURS [2.62]
TIME	N/A	=,>,>=,<,<=,<>	hh:mm, hh:mm:ss (Note: > and < make comparisons within a single day only) DAWN +/-mm DUSK +/-mm
DATE	N/A	=,>,>=,<,<=,<>	MM/DD/YYYY (Where YYYY is optional field, when completed comparisons are made within single year only)
DAY	N/A	=,<>	MON, TUE, WED, THU, FRI, SAT, SUN, HOL
MONTH	N/A	=,<>	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
COUNTER	####	=,>,>=,<,<=,<>	# (up to 10 digits representing counter value)
TIMER	####	=,>,>=,<,<=,<>	hh:mm:ss, hh:mm
POWERUP			
ALARM			

COMMANDS

The following is a list of commands. These belong between the THEN, ELSE, ELSE IF, and ENDIF statements. The action can also be another object, in which case the value of the second object is used to set the first object. Note that [] indicate optional parameters. Since version 2.63, an optional priority and fade can be added after the level (note that fade for group is ignored since the group has a fade with each member).

OBJECT	ID	ACTION
[DEVICE #] DIMMER [DEVICE #] RELAY [DEVICE #] PARTITION [DEVICE #] SWITCH [DEVICE #] ANALOG [DEVICE #] BUTTON PHONE DMX [DEVICE #] LEGACY-DIMMER [DEVICE #] DALI-DIMMER [DEVICE #] DALI-GROUP [DEVICE #] BINARY-OUTPUT	####	= 0-100 [@ 1-16] [FADE hh:mm or hh:mm:ss] += 0-100 -= 0-100 = DIMMER #### = RELAY #### = PARTITION #### = SWITCH #### = ANALOG #### = BUTTON #### = PHONE #### = DMX #### = LEGACY-DIMMER #### ON [@ 1-16] [FADE hh:mm or hh:mm:ss] OFF [@ 1-16] [FADE hh:mm or hh:mm:ss] WARN [@ 1-16] [FADE hh:mm or hh:mm:ss] STOP [@ 1-16] [FADE hh:mm or hh:mm:ss] LOWER [@ 1-16] [FADE hh:mm or hh:mm:ss] RAISE [@ 1-16] [FADE hh:mm or hh:mm:ss] NULL [@ 1-16] [FADE hh:mm or hh:mm:ss] ENABLE [ON,OFF,TIMEOUT,WARN] DISABLE [ON,OFF,TIMEOUT,WARN] ~~PRIORITY = 1-16~~
DIMMER [v2.23] RELAY	####	RESET STARTS RESET HOURS ~~PRIORITY = 1-16~~
GROUP	####	= 0-100 [@ 1-16] += 0-100 -= 0-100 ON [@ 1-16] OFF [@ 1-16] WARN [@ 1-16] STOP [@ 1-16] LOWER [@ 1-16] RAISE [@ 1-16] NULL [@ 1-16] ENABLE [ON,OFF] DISABLE [ON,OFF] ~~PRIORITY = 1-16~~
COUNTER	####	RESET [#] INC [#] DEC [#] ENABLE DISABLE
TIMER	####	RESET [hh:mm:ss or hh:mm] START STOP ENABLE DISABLE
SCHEDULE		## CLEAR [2.17]
MESSAGE note: v2.62 logs this message of up to 16 characters.		“User Text” CLEAR

DEBUGGING

The Synergy scripting engine has debugging built in, and can be enabled and monitored using the serial console. The DEBUG command displays the debug information to the serial console. The DEBUG OFF command displays the debug information to the VGA card, if present. The LOGIC DEBUG command turns on the script logic debug information. The LOGIC DEBUG OFF command turns off the script logic debug information.

The LOGIC command will parse and add the script commands to the system from the logic.txt file. The LOGIC CLEAR command will remove any logic data from memory. The LOGIC FILENAME.TXT will parse a file named FILENAME.TXT, and FILENAME.TXT can be any filename.

The GRP command can be used to view the logic groups that are in the system. The group numbers 10000 and above are logic groups. Using the command GRP 10013 will display the information about the logic group 10013, which was formed from a logic statement or expression. Information about other members of the script logic can be viewed using the GRP command, the TRG command, and the ACT command.

For example, the following script logic:

```
IF (SWITCH 102 OFF AND SWITCH 0103 OFF) AND SWITCH 0201 OFF THEN
  RELAY 105 OFF
ENDIF
```

is loaded into the system and can be viewed from the serial console as follows:

```
>grp 10013
  GROUP 10013
  "Logic Group"
  Group Level:   255
  Enable ON/OFF: Y/Y
  Triggers:
    *L-Term 10014 (IF LEVEL SWITCH 102 = 0)
    *L-Term 10015 (IF LEVEL SWITCH 103 = 0)
    *L-Term 10017 (IF LEVEL SWITCH 201 = 0)
    *L-Expr 10016 (L-Term 10015 AND L-Term 10014)
    +*L-Expr 10018 (L-Term 10017 AND L-Expr 10016)
  Actions:
    ACTION 10019: THEN TO_LEVEL 0 to RELAY 105 (0)
```

The asterisk "*" beside a trigger means that it has evaluated as TRUE or active. The plus sign "+" beside a trigger means that the trigger is the master trigger.

The script logic file can be viewed using the TYPE command: TYPE LOGIC.TXT

EXAMPLES

The following are examples on how to do various things using the script logic.

ADDING TIMEOUT TO A SWITCH

This script logic adds a timeout to a SWITCH by redirecting it from SWITCH 0101 to virtual SWITCH 1501 between the hours of 5:30 PM and midnight. Note that SWITCH 1501 needs to be configured with a timeout and should control the same GROUP.

```
IF SWITCH 101 ON AND TIME > 17:30 THEN
  SWITCH 1501 ON
ENDIF
```

A-B-C BALLROOM

This logic changes the switch to group relationship for SWITCH 201 based upon a combination of other inputs.

```
IF (SWITCH 102 OFF AND SWITCH 103 OFF) AND SWITCH 201 ON THEN
  GROUP 105 ON
ENDIF
IF (SWITCH 102 OFF AND SWITCH 103 OFF) AND SWITCH 201 OFF THEN
  GROUP 105 OFF
ENDIF

IF (SWITCH 102 ON AND SWITCH 103 OFF) AND SWITCH 201 ON THEN
  GROUP 106 ON
ENDIF
IF (SWITCH 102 ON AND SWITCH 103 OFF) AND SWITCH 201 OFF THEN
  GROUP 106 OFF
ENDIF

IF (SWITCH 102 OFF AND SWITCH 103 ON) AND SWITCH 201 ON THEN
  GROUP 107 ON
ENDIF
IF (SWITCH 102 OFF AND SWITCH 103 ON) AND SWITCH 201 OFF THEN
  GROUP 107 OFF
ENDIF

IF (SWITCH 102 ON AND SWITCH 103 ON) AND SWITCH 201 ON THEN
  GROUP 105 ON
  GROUP 106 ON
  GROUP 107 ON
ENDIF
IF (SWITCH 102 ON AND SWITCH 103 ON) AND SWITCH 201 OFF THEN
  GROUP 105 OFF
  GROUP 106 OFF
  GROUP 107 OFF
ENDIF
```

DISABLE AN INPUT DURING THE DAY

This logic disables the off action of a switch during the hours of 9 AM to 5 PM. Redirecting inputs as shown in a previous example could also do this. Note that TIME and DATE expressions will not be in effect until the scheduler starts (about 30 seconds after the last module comes online).

```
IF TIME > 9:00 AND TIME < 17:00 THEN
  SWITCH 101 DISABLE OFF
ELSE
  SWITCH 101 ENABLE
ENDIF
```

Triplexing an Input

This logic creates a triplexed input. Each time the switch is pressed, a different action takes place until the counter resets. This can be used for as many states are necessary, not just 3 states. The RESET should always be done when the counter exceeds a value (use greater than).

```
IF SWITCH 101 ON THEN
  COUNTER 101 INC 1
ENDIF

IF COUNTER 101 = 1 THEN
  GROUP 101 = 33
ENDIF

IF COUNTER 101 = 2 THEN
  GROUP 101 = 66
ENDIF

IF COUNTER 101 = 3 THEN
  GROUP 101 = 100
ENDIF

IF COUNTER 101 > 3 THEN
  GROUP 101 = 0
  COUNTER 101 RESET 0
ENDIF
```

Disable Analog Input During Weekend

This script logic disables all events from turning ON GROUP 0101 on Saturdays and Sundays (Used to disable ON event for analog inputs over the weekend).

```
IF DAY = SUN OR DAY = SAT THEN
  GROUP 101 DISABLE ON
ELSE
  GROUP 101 ENABLE
ENDIF
```

Disable Sensor During the Day

This disables exterior occupancy sensor from turning the lights ON during daytime hours.

```
IF TIME > DAWN-30 AND TIME < DUSK+15 THEN
  ANALOG 101 DISABLE ON
ELSE
  ANALOG 101 ENABLE
ENDIF
```

Keeping a Hallway Lit When Other Lights are On

This script logic keeps path of egress relay 0104 ON if other relays in the area are ON, and times them out when all other relays are turned OFF, through the use of a virtual switch, SWITCH 104. Note that SWITCH 104 must be configured with a timeout and control a group that controls the desired relay.

```
IF RELAY 101 ON OR RELAY 102 ON OR RELAY 103 ON THEN
  RELAY 104 ON
ENDIF

IF RELAY 101 OFF AND RELAY 102 OFF AND RELAY 103 OFF THEN
  SWITCH 104 ON //starts timeout
ENDIF
```

Virtual Switch Using Phone

This script logic adds a timeout to an input by using virtual SWITCH 1501 between the hours of 5:30 PM and midnight.

```
IF PHONE 101 > 85 AND TIME > 17:30 THEN
  SWITCH 1501 ON
ENDIF
```

Timeout and Warn

This is a representation of a group timing out over a 2-1/2 hour period with single output warn at 15:00 before OFF not using a virtual switch input.

```
IF SWITCH 101 ON AND TIME > 17:30 THEN
  GROUP 101 ON
  TIMER 101 RESET
  TIMER 101 START
ENDIF

IF TIMER 101 = 02:15:00 THEN
  GROUP 101 WARN
ENDIF

IF TIMER 101 = 02:30:00 THEN
  GROUP 101 OFF
ENDIF
```

Dynamic Schedule Changes

This script logic runs different schedules for Mondays in the month of June.

```
IF MONTH = JUN AND DAY = MON THEN
  SCHEDULE 5
ENDIF
```

Lamp Lumen Depreciation Compensation

This sets maximum high end output allowing dimmers to compensate for lamp lumen depreciation based on dimmer run time.

```
IF RELAY 1501 ON AND RELAY 1501 HOURS >= 7000 THEN
  GROUP 0201 95
ENDIF

IF RELAY 1501 ON AND RELAY 1501 HOURS < 7000 THEN
  GROUP 0201 90
ENDIF

IF RELAY 1501 ON AND RELAY 1501 HOURS < 4000 THEN
  GROUP 0201 85
ENDIF

IF RELAY 1501 ON AND RELAY 1501 HOURS < 1000 THEN
  GROUP 0201 80
ENDIF

IF RELAY 1501 OFF THEN
  GROUP 0201 OFF
ENDIF
```

User Interface Message for Lamp Change

This displays a note on the UIP advising that lamps should be changed.

```
IF RELAY 0101 STARTS > 3000 OR RELAY 0101 HOURS > 15000 THEN
  MESSAGE "RELAMP OUTPUT 0101"
  RELAY 0101 RESET HOURS
  RELAY 0101 RESET STARTS
ENDIF

// use a switch input to clear the message
IF SWITCH 101 ON THEN
  MESSAGE CLEAR
ENDIF
```

User Interface Message Display

This script logic counts the number of switch operations, and displays UIP message. This script logic does not effect the switch controlling the normal group, and thus, does not require additional statement in script to make sure the group turns ON. Commands don't need to be ANDed – just place them after the THEN or ELSE and before ENDIF.

```
IF SWITCH 101 ON AND TIME > 17:30 THEN
  COUNTER 101 INCREMENT 1
ENDIF

IF DAY = MONDAY THEN
  COUNTER 102 INCREMENT 1
ENDIF

IF COUNTER 102 > 10 THEN
  COUNTER 102 RESET 0
  COUNTER 101 RESET 0
  MESSAGE CLEAR
ENDIF

IF COUNTER 0101 > 10 THEN
  MESSAGE "ADJUST SCHEDULE FOR OFFICE AREA"
  //truncates if > 20 characters
ENDIF
```

Occupancy Sensor Controlled Adjacent Zones

This script logic enables and disables groups such that each zone and its adjacent zones are always lighted when the sensor for that zone indicates occupancy. Note that the switch input (occupancy sensor) must be programmed separately to control the group (zone).

```
// EDGE OF FLOOR
IF SWITCH 101 ON THEN
  GROUP 101 DISABLE OFF
  GROUP 102 DISABLE OFF
ENDIF

// IF ADJACENT ZONES AND MY ZONE IS OFF, THEN I CAN GO OFF
IF SWITCH 101 OFF AND SWITCH 102 OFF THEN
  GROUP 101 ENABLE
  GROUP 101 OFF
ENDIF

// NEXT ZONE
IF SWITCH 102 ON THEN
  GROUP 101 DISABLE OFF
  GROUP 102 DISABLE OFF
  GROUP 103 DISABLE OFF
ENDIF

// IF ADJACENT ZONES AND MY ZONE IS OFF, THEN I CAN GO OFF
IF SWITCH 101 OFF AND SWITCH 102 OFF AND SWITCH 103 OFF THEN
  GROUP 102 ENABLE
  GROUP 102 OFF
ENDIF

// NEXT ZONE
IF SWITCH 103 ON THEN
  GROUP 102 DISABLE OFF
  GROUP 103 DISABLE OFF
  GROUP 104 DISABLE OFF
ENDIF

// IF ADJACENT ZONES AND MY ZONE IS OFF, THEN I CAN GO OFF
IF SWITCH 102 OFF AND SWITCH 103 OFF AND SWITCH 104 OFF THEN
  GROUP 103 ENABLE
  GROUP 103 OFF
ENDIF
```

Disable Occupancy Sensor During the Day, Except Weekends

This script logic disables an occupancy sensor input during the day, except on weekends and holidays. Note that the switch input (occupancy sensor) must be programmed separately to control the desired group.

```
IF DAY <> SAT AND DAY <> SUN AND DAY <> HOL AND
   TIME > DAWN AND TIME < DUSK THEN
  SWITCH 101 DISABLE
ELSE
  SWITCH 101 ENABLE
ENDIF
```

Room Assignment - 2 Rooms

This script logic assigns a switch to control multiple groups when a partition (virtual relay) is ON. Note that the SWITCH in the first IF is used to trigger the event - all SWITCH 101 events will be used. Also note that you must program SWITCH 101 and SWITCH 102 to control GROUP 101, which contains both rooms.

```
// allow room assignment to survive after powerup
IF POWERUP AND RELAY 701 OFF THEN
  GROUP 101 DISABLE
ENDIF

IF POWERUP AND RELAY 701 ON THEN
  GROUP 101 ENABLE
ENDIF

IF RELAY 701 ON THEN
  GROUP 101 ENABLE
ENDIF

IF RELAY 701 OFF THEN
  GROUP 101 DISABLE
ENDIF

IF SWITCH 101 >=0 THEN
  GROUP 102 = SWITCH 101
ENDIF
IF SWITCH 102 >=0 THEN
  GROUP 103 = SWITCH 102
ENDIF
```

Room Assignment - 3 Rooms

This script logic uses 2 analog inputs to determine which partition is closed. Note that the "SWITCH >= 0" is used to trigger the event on any level. Also note that you must program the switches to control the combination of room 1 + 2 + 3. We assume that room 2 is in the middle.

```
// POSSIBLE COMBINATIONS IF ROOM 2 IS IN THE MIDDLE
// GROUP 301 - ROOM 1 + 2 + 3
// GROUP 302 - ROOM 1 + 2
// GROUP 303 - ROOM 2 + 3
// GROUP 304 - ROOM 1
// GROUP 305 - ROOM 2
// GROUP 306 - ROOM 3

// allow the logic to survive a power down
// by re-checking the analog key-switch
// assume that all are enabled at powerup
IF POWERUP THEN
// ROOM 1 + 2 + 3
IF ANALOG 101 OFF AND ANALOG 102 OFF THEN
  GROUP 301 ENABLE
  GROUP 302 DISABLE
  GROUP 303 DISABLE
  GROUP 304 DISABLE
  GROUP 305 DISABLE
  GROUP 306 DISABLE
ENDIF
// ROOM 1 + 2
// ROOM 3
IF ANALOG 101 OFF AND ANALOG 102 ON THEN
  GROUP 301 DISABLE
  GROUP 302 ENABLE
  GROUP 303 DISABLE
```

```
  GROUP 304 DISABLE
  GROUP 305 DISABLE
  GROUP 306 ENABLE
ENDIF
// ROOM 1
// ROOM 2 + 3
IF ANALOG 101 ON AND ANALOG 102 OFF THEN
  GROUP 301 DISABLE
  GROUP 302 DISABLE
  GROUP 303 ENABLE
  GROUP 304 ENABLE
  GROUP 305 DISABLE
  GROUP 306 DISABLE
ENDIF
// ROOM 1
// ROOM 2
// ROOM 3
IF ANALOG 101 ON AND ANALOG 102 ON THEN
  GROUP 301 DISABLE
  GROUP 302 DISABLE
  GROUP 303 DISABLE
  GROUP 304 ENABLE
  GROUP 305 ENABLE
  GROUP 306 ENABLE
ENDIF
ENDIF

// key-switch enables the groups containing both rooms
// NOTE: BOTH SWITCHES MUST BE PROGRAMMED TO CONTROL THE
// SAME GROUP FOR THIS SCHEME TO WORK
// ALSO, THE ANALOG SHOULD NOT CONTROL ANY GROUPS
// ROOM 1 + 2 + 3
IF ANALOG 101 OFF AND ANALOG 102 OFF THEN
  GROUP 301 ENABLE
  GROUP 302 DISABLE
  GROUP 303 DISABLE
  GROUP 304 DISABLE
  GROUP 305 DISABLE
  GROUP 306 DISABLE
ENDIF
// ROOM 1 + 2
// ROOM 3
IF ANALOG 101 OFF AND ANALOG 102 ON THEN
  GROUP 301 DISABLE
  GROUP 302 ENABLE
  GROUP 303 DISABLE
  GROUP 304 DISABLE
  GROUP 305 DISABLE
  GROUP 306 ENABLE
ENDIF
// ROOM 1
// ROOM 2 + 3
IF ANALOG 101 ON AND ANALOG 102 OFF THEN
  GROUP 301 DISABLE
  GROUP 302 DISABLE
  GROUP 303 ENABLE
  GROUP 304 ENABLE
  GROUP 305 DISABLE
  GROUP 306 DISABLE
ENDIF
// ROOM 1
// ROOM 2
// ROOM 3
IF ANALOG 101 ON AND ANALOG 102 ON THEN
  GROUP 301 DISABLE
  GROUP 302 DISABLE
  GROUP 303 DISABLE
```

```
  GROUP 304 ENABLE
  GROUP 305 ENABLE
  GROUP 306 ENABLE
ENDIF

// allows the activation of the split rooms
// note that we didn't include duplicates
// ROOM 1 + 2
IF SWITCH 0101 >= 0 THEN
  GROUP 302 = SWITCH 101
ENDIF
IF SWITCH 0102 >= 0 THEN
  GROUP 302 = SWITCH 102
ENDIF
// ROOM 3
IF SWITCH 0103 >= 0 THEN
  GROUP 306 = SWITCH 103
ENDIF
// ROOM 1
IF SWITCH 0101 >= 0 THEN
  GROUP 304 = SWITCH 101
ENDIF
// ROOM 2 + 3
IF SWITCH 0102 >= 0 THEN
  GROUP 303 = SWITCH 102
ENDIF
IF SWITCH 0103 >= 0 THEN
  GROUP 303 = SWITCH 103
ENDIF
// ROOM 2
IF SWITCH 0102 >= 0 THEN
  GROUP 305 = SWITCH 102
ENDIF
```

Level Limiting

This script logic uses the value of an analog input as a limiter and limits the output as necessary.

```
IF DIMMER 101 > ANALOG 101 THEN
  DIMMER 101 = ANALOG 101
ENDIF
```

Controller Power Up

This script logic uses the POWERUP to perform actions when the controller first starts after a power-down sequence. This could be used for HID lamps containing dual controls. Note that the POWERUP expression cannot contain any nested IF expressions. Use AND or OR and multiple POWER statements to facilitate the lack of nested IF expressions.

```
IF POWERUP THEN
  RELAY 101 OFF
  RELAY 103 OFF
  RELAY 105 OFF
  RELAY 107 OFF
ENDIF
```

HID Warn Before OFF

This logic script links the power relays to the horn relay: whenever the horn relay is sent an ON command, the HID power relays will be turned ON but the alarm will remain silent (Note that the horn relay should be programmed as Momentary OFF type and will only close on an

OFF signal). When the horn relay is sent an OFF command, the horn will sound momentarily and the timer starts. If no ON command are issued during the next ten minutes, the HID power relays are then turned OFF.

The advantage of programming it this way is that there are no limitations or special conditions imposed on how these lights are programmed. In other words, the switches that will control the lights can have timeouts, priorities can be used as normal, etc.

In this example, GROUP 1 contains the HID power relay(s). RELAY 102 is the Horn warning relay and is set up as Momentary OFF type through the controller. GROUP 102 contains Relay 102 only. All schedules, switches, and stations that control the HID lights for this area are programmed to control GROUP 102. The following logic script is then used:

```
IF RELAY 102 OFF THEN
  TIMER 1 RESET
  TIMER 1 START
ELSE
  TIMER 1 RESET
  GROUP 1 ON
ENDIF

IF TIMER 1 > 00:10:00 THEN
  GROUP 1 OFF
ENDIF
```

Signal an Alarm

This logic script turns on a relay when an alarm is active on the user interface panel.

```
IF ALARM THEN
  RELAY 101 ON
ELSE
  RELAY 101 OFF
```

Even Out the Running Hours or Strikes

This logic script turns on the relay with the least number of hours or strikes in response to a switch press. Note that we check for both relays OFF state to keep the logic from acting on the second event (the relay turning on) through the queue. Note that the script is no longer case sensitive.

```
if switch 101 on then
  if relay 101 off and relay 102 off then
    if relay 101 starts > relay 102 starts then
      relay 102 on
    else
      relay 101 on
    endif
  endif
endif

if switch 101 off then
  relay 101 off
  relay 102 off
endif
```

Switch Input Timeout Masking

This logic script disables the warning blink normally created by a switch input timeout during occupied times of the day.

```
IF DAY <> SAT AND DAY <> SUN AND TIME > 5:00 AND TIME < 17:00 THEN
  SWITCH 101 DISABLE WARN
ELSE
  SWITCH 101 ENABLE WARN
ENDIF
```

Remote Object Commands

This logic script disables the timeout normally created by a switch input during occupied times of the day.

```
IF SWITCH 101 ON THEN
  DEVICE 127 SWITCH 101 ON
  DEVICE 101 ANALOG 101 = 100
ENDIF
```

Simulating a Command Object for Scenes

This logic script uses a dimmer to simulate a command object to create unique scenes.

```
IF DIMMER 1 = 1 THEN
  GROUP 101 ON
ENDIF

IF DIMMER 1 = 2 THEN
  GROUP 102 ON
ENDIF

IF DIMMER 1 = 3 THEN
  GROUP 103 ON
ENDIF

IF DIMMER 1 = 4 THEN
  GROUP 101 OFF
ENDIF
```