WAKE TECH
Building, Engineering, & Skilled Technologies

Logic with Switch Blocks

BAT-212: BAT Logic and Programming

This material is based upon work supported by the National Science Foundation Advanced Technical Education grant program, A New Technician Training Program for Advanced Building Technologies, DUE-2000190.

The opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Logic with Switch Blocks

OBJECTIVES

Upon completion of this activity the student will be able to:

1. Use Sedona basic logic blocks to implement Boolean combinational logic.
2. Use Sedona switch blocks to implement conditional logic.

PARTS AND EQUIPMENT

- Sedona Application Editor from Contemporary Controls
- BAS22controller or BASemulator
- BAT trainer

REFERENCES

- Contemporary Controls product page: https://www.ccontrols.com/basautomation
- Sedona Open Control Reference Manual: Course Resources or https://www.ccontrols.com/pdf/RM-SEDONA00.pdf

BACKGROUND

The basic logic gates can be combined to form more complicated logic systems. Combinational logic refers to combinations of the basic gates and has the property that any change in input affects the output immediately, allowing for propagation delay. This contrasts to sequential logic, where the effect of a change in input depends on the current state of the system.

Switches allow for logic to be applied to analog signals, selecting which analog signal passes through the switch based on selector bits. (Sedona Ref. Manual 2.5) Also, comparators allow generation of binary values from analog values by applying a threshold to the analog value (Sedona Ref. Manual 3.24)

Latches allow one value to set and other value to reset, so that a response to a particular input depends on the previous state. (Sedona Ref. Manual 2.3).

PROCEDURES

Part 1:

The scenario is control for a room lighting. The inputs for control will be 2 binary sensors - occupancy true/false, override true/false – and an analog ambient light sensor that is compared to two setpoints to determine if light levels are too high or too low. The output control will be 1 binary output: lights on/off, or shades open/close. Generate a truth table and a logic schematic for each output for the following specification, using the setpoint comparison outputs as binary

values. Then use the logic schematic as a starting point for the wiresheet program for each specification.

a. Light on/off: Light should be on if ambient light is too low and occupied, or if override is true.

b. Shades open/close: If ambient light is too high, close the window shades. The shades should be closed if override is true.

Use the webpage of the controller to force inputs and display outputs instead of connecting to physical I/O in order to test your wiresheet. You can work on the controller or the emulator. (If you work on the controller, it will be easy to wire to the trainer after testing with the webpage.)

Demonstrate your working programs. Verify that your predicted results in the truth tables are the actual results of your circuits. Include tables, logic schematics, wiresheet images and photo of the circuit in your lab report.

Part 2:

The scenario is to use set parameters for calculating setpoint and deadband temperatures for a heat only system. Input parameters will come from web components via the controller webpage. Usually these values would come to the controller from a supervisory controller or other high level control system. These web component input parameters are: 1) heating setpoint; 2) deadband; and 3) unoccupied offset. Input sensors are an analog temperature sensor to measure room air temperature (RAT) - thermistor on the trainer- and an occupancy sensor (use one of the toggle switches). The output of your wiresheet will be the active temperature setpoint, which should be sent to a web component, and a binary indicator (LED) showing when heat is on.

The logic to be implemented is:

1. If occupied, then the heat setpoint is as given by the web component.
2. If unoccupied, subtract the unoccupied offset from the heat setpoint.
3. The result from the previous two steps is the active heat setpoint.
4. If temperature is lower than the active heat setpoint, then heat turns on and remains on until temperature is greater than setpoint plus deadband.

Test your emulation with the following data:

Heat setpoint: 68 deg F Unoccupied offset: 3 deg F Deadband: 2 deg F

Then test with the following room air temperatures and occupancy settings:

RAT	Occupied	Heat on	Active setpoint
74	false		
67	true		
65	false		
63	true		
62	false		

Demonstrate your working wiring sheet to the instructor. Include table and screen shots of wiresheets in your lab report.

Backup your wiresheet using the BASbackup tool, as you will use this as a starting point for the next lab.