

Sensing Switch Movement

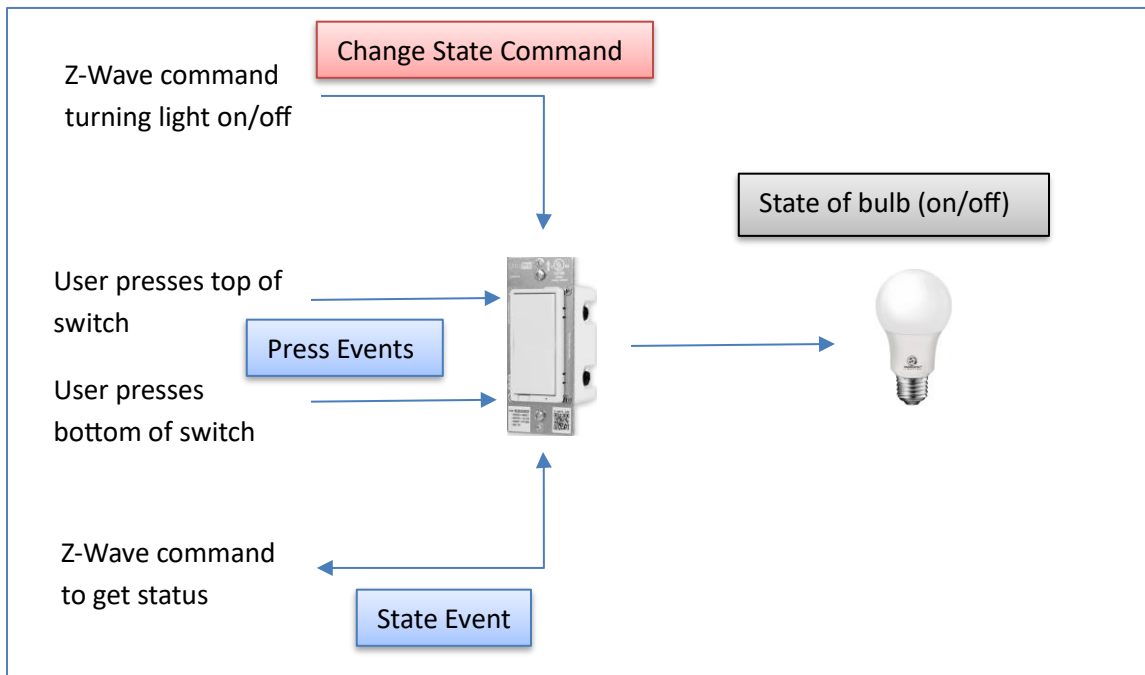
It is sometimes useful to be able to tell if a user has physically pressed a light switch. An example of this would be as follows:

- Home controller detects garage door opening and automatically turns on garage lights for 3 minutes, and then turns them off. This works great when you have to go in the garage to take the trash out and don't want to bother turning the light on/off.
- However, sometimes you want to stay in the garage longer, and need to override the home controller logic that automatically turns the light off. In this case it is handy to simply press the top of the garage light switch to tell the home controller to override its default behavior and leave the light on.

This takes advantage of the following Z-Wave behavior.

In a light switch, the state of the bulb can be changed by either:

1. A change state command issued by the home controller
2. A user physically pressing the top/bottom of the paddle switch



Sensing the State of the Bulb

When either case occurs, a “State Event” is sent to the home controller which can trigger various logic. However, it is impossible to tell from the “State Event” which set of events caused the change of state. The following ISY logic illustrates how to trigger on the “State Event” using the “Status is On/Off” event trapping.

```
Program Content for 'Run Circulation Pump'  
If  
    'Master Bath Light' Status is On  
    Or 'Spare Bath Light' Status is On  
Then  
    Set 'Hot Water Pump' On  
    Wait 3 minutes  
    Set 'Hot Water Pump' Off  
Else  
    - No Actions - (To add one, press 'Action')  
  
the circulation pump only needs to be turned on long enough to get  
water to the end of the pipe loop.
```

Sensing when the User Physically Presses the Switch

When the user physically presses the switch, a “Press Event” is sent to the home controller in addition to the “State Event” described above. This allows the logic to differentiate whether the light was turned on due to a home controller programmatic change, or because a physical person actually pressed the switch.

IMPORTANT: the “is switched on/off” test is accessed through the “Control” (not “Status”) node component.

Depending on the device AND the home controller Z-Wave implementation, you will need to use different logic to distinguish user button presses.

Older Polisy Implementation

Because the older Polisy Z-Wave implementation was somewhat limited, it did not have support for scene controllers (which is also why the scene controller buttons didn’t even appear on devices supporting them.)

So the on-off paddle switch was described as a single on-off switch and you can use the following logic to sense this “Press Event” as follows using the “is switched on/off” event trapping.

```
Program Content for 'Garage Light Switch Off'
If
    'Garage Ceiling Lights' is switched Off
Then
    Wait 30 seconds
    $Garage_Light_Override = 0
Else
    - No Actions - (To add one, press 'Action')

Using 'Condition' 'Control' on a light switch and checking for 'off' (which gets translated to 'is switched off')
only becomes active when the switch is physically pressed, and NOT when the switch changes state programmatically.
This is useful to provide a way to 'override' programmatic behavior for a switched circuit without having to add
another Z-Wave button in the area.

wait here allows the user to be able to exit the garage without the program turning the lights back on again
```

Newer Eisy Implementation

The eISY Z-Wave implementation is much more advanced and supports extra nodes that report as scene buttons. It is these scene buttons, and not the main switch node, that must be monitored for user presses.

```
Program Content for 'Button Test'
If
    'Garage Ceiling Lights / ZY 035 Scene Button 1' is switched On
    Or 'Garage Ceiling Lights / ZY 035 Scene Button 2' is switched On
Then
    Send Notification to 'default' content 'Device Status Changed'
Else
    - No Actions - (To add one, press 'Action')
```