



HCA Tech Note 710

ZWave Keypads

ZWave keypads that interface through Hubitat are similar in form and function to Insteon and UPB keypads. However, the way that you construct program triggers for them – what program starts on which button is pressed – is done differently. This technical note explains how to do it.

While there are many ZWave keypads available, I have worked with this one so will use it in my examples.

<https://www.amazon.com/Z-Wave-Network-Scene-Controller-Required/dp/B09B6S4TSL>

(As always, I link to Amazon because it is easy to find these kinds of products. I'm sure they are available elsewhere. If you buy them from Amazon or elsewhere, I get no kickback)

It's a switch replacement so it has a load controller tied to the first button. Each button has a LED indicator, and you can set the color to be used (white, red, blue, or green) and what they do (on when button is activated and off when not activated, or the other way round). And lots of other nice features. It is Hubitat supported, and all the configuration settings can be adjusted there. But you can read about it on your own time, for now I want to talk about HCA support for it.

Actions that Hubitat reports – sensor value changes, a device is controlled, a keypad button is pressed – all arrive in the same manner. What the programs that receive these messages do is to update “tags” on the device to record the new data.

Let's take a minute to discuss the terms “tag” and “variable”. A variable is the term programmers give to a piece of state that has a name, a value, persistence, and visibility. Suppose a program has a “local” variable named “temperature” that has the value “68”. Since it is a “local” variable, that means that it only exists while the program is executing, and it is visible only to that program.

Now suppose your design has a global variable named “Humidity” with a value “86”. Since it is a global variable, it exists for as long as HCA is executing, and it is visible to all programs with that simple name.

“Tags” are like variables in that they have a name and a value. They are associated with a device, so they persist as long as that device exists. They are visible to any program that wants to see them but, unlike global variables where all a program needs to know is the name, the program must explicitly ask for them using expression functions that specify what object they are associated with.

Here is a screen capture of a sensor device in my home. There are many tags associated with this device.

TechSup@HCATech.com

www.homecontrolassistant.com



HCA Tech Note 710

"Hubitat Devices - Great Room Multi Sensor" Properties

Name	Notes	Room	Type	User Class Object	Options
Tag Name	Current Value				
HE_type	Aeon Multisensor 6				
HE_devType	Sensor				
HE_devSubType	motion				
HE_Humidity	56				
HE_Illuminance	192				
HE_Temperature	76.0				
Annotate					
HE_Acceleration	inactive				
HE_Battery	100				
HE_UltravioletIndex	0				
LastEvent	8/16/2022 10:29:45 AM				
HE_Motion	inactive				

In a program, if I want to get the temperature from that device, I would put this into a Compute element:

```
tempNow = _ObjectTagGet("Hubitat Devices-Great Room Multi Sensor", "HE_Temperature");
```

The advantage of "tags" over global variables is that the names need not be unique across your whole design. For example, one device can have a tag named "HE_Temperature" and another device can have the same named tag.

How does that relate to the keypad I mentioned above? Like sensors, each time a keypad button is pressed, HCA receives a message, and like the sensor example, it gets saved as a tag. Here is that keypad from my design:

TechSup@HCATech.com

www.homecontrolassistant.com



HCA Tech Note 710

"Hubitat Devices - Scene Keypad" Properties

Name	Notes	Room	Type	User Class Object
Tag Name	Current Value			
HE_type	Zooz Zen32 Scene Controller			
HE_devType	switch			
HE_devSubType	button			
HE_Pushed	4			
HE_Button_3_Trigg	8/7/2022 1:39:43 PM			
HE_Button_1_Trigg	8/7/2022 1:39:44 PM			
HE_Button_2_Trigg	8/7/2022 1:39:46 PM			
HE_Button_4_Trigg	8/7/2022 1:40:11 PM			
HE_DoubleTapped	1			
HE_Switch	off			
HE_Button_5_Trigg	8/7/2022 1:39:38 PM			

The button presses are recorded as “HE_Button_1_Trigger”, “HE_Button_2_Trigger”, etc. The tag value is a time stamp of when the message was received.

Now, how to use this to cause something to happen when a button is pressed?

Tags have another nice feature. There exists what is called a “Tag change trigger” for a program. What this trigger does is exactly what it says: When the named tag has its value changed, the program starts. Like this:



HCA Tech Note 710

Name	Notes	Triggers	Visual Programmer	Advanced Options	Restart	Icon	Di
------	-------	----------	-------------------	------------------	---------	------	----

Start the program on any of these triggers:

Trigger
▶ A change of tag "HE_Button_1_Trigger" on "Hubitat Devices - Scene Keypad"
▶ A change of tag "HE_Button_2_Trigger" on "Hubitat Devices - Scene Keypad"
▶ A change of tag "HE_Button_3_Trigger" on "Hubitat Devices - Scene Keypad"
▶ A change of tag "HE_Button_4_Trigger" on "Hubitat Devices - Scene Keypad"
▶ A change of tag "HE_Button_5_Trigger" on "Hubitat Devices - Scene Keypad"

This example program starts when any button is pressed – it has a trigger for each button on the keypad - and internally can test the “starting trigger” to decide which button was pressed.

(Tag change triggers are a bit more capable for the advanced programmer. The program is provided arguments that contain the object whose tag changed, the name of the tag, and the new tag value. In this example that wasn't needed because the trigger named the object and there was a trigger for each button. The program need only test on the starting trigger and it then knows what button was pressed.)

What's the takeaway from this? ZWave keypads exist, they are capable, they are Hubitat and HCA supported. To start programs when buttons are pressed a different technique is used than what was used for Insteon and UPB keypads.

##end##