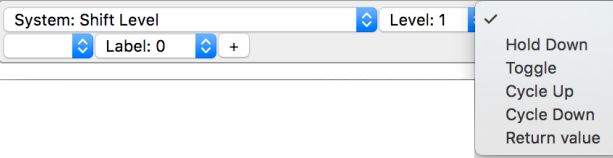
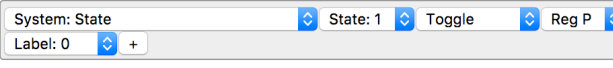
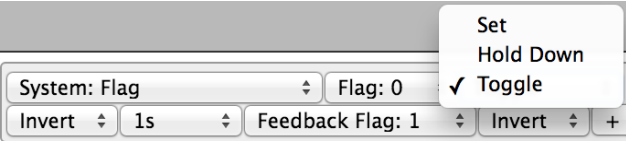
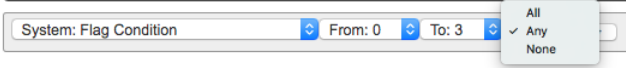
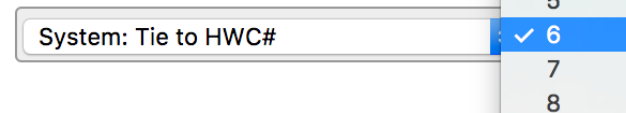


System Actions

This is a table of system actions available for any UniSketch powered SKAARHOJ controller:

<p>Shift Level</p> 	<p>Sets the controller shift level.</p> <p><i>Binary triggers:</i> Sets the selected shift level for the given register (default or A-D). If Hold Down is selected, the shift level will fall back to Level 0 whenever the trigger is released. Toggle will set the shift level unless it's the same in which case it will set Level 0. If Cycle Up/Down modes are selected, a trigger will set the next/previous shift level up to the level selected.</p> <p><i>Pulse inputs:</i> Will cycle through shift levels up to the selected level.</p> <p><i>Analog inputs:</i> Will map the analog input range to shift levels up to the selected level.</p> <p><i>Binary outputs:</i> On when the shift level matches selected source (or when trigger is held in Cycle modes)</p> <p><i>Button colors:</i> Follows binary output: Highlighted, when on.</p> <p>Mode "Return value" means the selected level will be set if the binary value of the previous actions return value is true, otherwise the level will be set to zero. This is evaluated regardless of any triggers. This feature is a way to link a state of an external device to a shift level in the controller.</p> <p><i>Displays:</i> Shows the selected shift level either as a number or "Off" (0) or "On" (1). If Cycle modes are selected, the display will reflect the current level (status). If a Label is selected other than "Label: 0" either a label string or an image (1-10) is used. Label strings will only overlay the default part (header, line1, line 2) if they are not empty.</p>
<p>State</p> 	<p>Sets the controller state.</p> <p><i>Binary triggers:</i> Sets the selected state. If Hold Down is selected, the state will fall back State 0 whenever the trigger is released. Toggle will set the state unless its the same in which case it will set State 0. If Cycle Up/Down modes are selected, a trigger will set the next/previous state up to the state number selected. If Clear/Cycle Current Shift is selected, the currently active shift register (in the section/module where this HWc is located) will be reset in case the state changes from another state, alternatively if the state is already equal to the value set by the trigger, the shift register will be cycled up to the value selected for "Cycle Shift" and then fall over to zero again.</p> <p><i>Pulse inputs:</i> Will cycle through states up to the selected state number. If Clear/Cycle Current Shift is selected, the currently active shift register will be reset to zero.</p> <p><i>Analog inputs:</i> Will map the analog input range to state numbers up to the selected state.</p> <p><i>Binary outputs:</i> On when the controller state matches the selected state (or when trigger is held in Cycle modes)</p> <p><i>Button colors:</i> Follows binary output: Highlighted, when on.</p> <p><i>Display text:</i> For displays and smart switches, the value will be shown as the preset label entered in the web interface if given, otherwise as a number. If a Label is selected other than "Label: 0" either a label string or an image (1-10) is used. Label strings will only overlay the default part (header, line1, line 2) if they are not empty.</p>

<div> <div>Memory</div> <div> System: Memory K 5 Toggle Persist INS CP </div> <div> Image #1 + </div> </div>	<p>Sets memory registers A-L. If "Persist" is selected, the value will be stored in EEPROM and recalled from EEPROM upon booting the controller again (only register A-D). Values set in memory group A and B will also be set in Memory Group A and B.</p> <p><i>Binary triggers:</i> Sets the selected value for the given register A-D. If Hold Down is selected, the value will fall back to the previous value whenever the trigger is released. Toggle will set the value, but on a subsequent trigger, it will fall back to the previous value. If Cycle Up/Down modes are selected, a trigger will set the next/previous value up to the value selected.</p> <p><i>Pulse inputs:</i> Will cycle through values up to the selected value.</p> <p><i>Analog inputs:</i> Will map the analog input range to values up to the selected value.</p> <p><i>Binary outputs:</i> On when the memory value matches the selected value (or when trigger is held in Cycle modes)</p> <p><i>Button colors:</i> Follows binary output: Highlighted, when on.</p> <p><i>Display text:</i> For displays and smart switches, the value will be shown as a number. If a Label is selected other than "Label: 0" either a label string or an image (1-10) is used instead. Label strings will only overlay the default part (header, line1, line 2) if they are not empty.</p>
<div> <div>Cycle Memory</div> <div> System: Cycle Memory A From: 4 To: 1 </div> <div> From: 20 To: 22 And: 0 Persist </div> </div>	<p>Sets memory registers A-D with values from specified ranges. If "Persist" is selected, the value will be stored in EEPROM and recalled from EEPROM upon booting the controller again. You can define two From-To ranges (values inclusive) and a single value ("And", must be different from zero). If both values in the range definitions are zero, the range is ignored.</p> <p><i>Binary triggers:</i> Cycles to the next value given by the range line up.</p> <p><i>Pulse inputs:</i> Will cycle through the values in the range line up.</p> <p><i>Analog inputs:</i> Will map the analog input range to values in the range line up.</p> <p><i>Binary outputs:</i> On when trigger is held in Cycle modes.</p> <p><i>Button colors:</i> Follows binary output: Highlighted, when on.</p> <p><i>Displays:</i> Shows the memory value.</p>

<p>Flag</p> 	<p>Sets a memory flag which is an internal binary value. This can be used to transport binary values around in the system.</p> <p><i>Binary triggers:</i> Sets the selected flag. If Hold Down is selected, the flag will fall back to the previous value whenever the trigger is released. Toggle will set the flag, but on a subsequent trigger, it will fall back to the previous value.</p> <p><i>Pulse inputs:</i> Will flip the value</p> <p><i>Analog inputs:</i> Will clear/set the value when on either side of the middle of the analog value range.</p> <p><i>Binary outputs:</i> On, if the selected feedback flag is set (but subject to modification by the second invert option). Notice that you must select the same feedback flag number as the flag-number if you want it to respond "intuitively". The idea of the feedback flag is to have a way to send a value out of the system but only reflect a confirmative return value.</p> <p><i>Button colors:</i> Follows binary output: Highlighted, when on</p> <p><i>Other features:</i> If "Invert" is selected, the flag will be cleared (binary zero) instead of set (binary 1) in any of the above cases. If any time frame is set, the value will fall back to the default after that period of time.</p> <p><i>Display text:</i> For displays and smart switches, the status or function will be shown as On or Off (status us used for toggle switches). If a Label is selected other than "Label: 0" either a label string or an image (1-10) is used instead. Label strings will only overlay the default part (header, line1, line 2) if they are not empty.</p>
<p>Flag Condition</p> 	<p>Output only action.</p> <p>Will output binary true if the range of system flags between from and to (both included) satisfies the condition set in the third drop down. The options are:</p> <ul style="list-style-type: none"> - All: Will return true if all the flags are set, otherwise false - Any: Will return true if any of the flags are set, otherwise false - None: Will return true if none of the flags are set, otherwise false <p>This action can be used together with GPI outputs, or as an input to "Synthesized Trigger" on a Virtual Hardware Component to create complex behaviours.</p>
<p>Tie to HWC#</p> 	<p>This will tie an interface component to another interface component given by it's "Hwc#" number which is the number found in the web interface on every component. This feature is useful if you want a display to be linked to a button to display a label for it. But you can also copy functionality otherwise.</p> <p>If you tie a HWC to another HWC, you will execute its behavioural configuration, but not necessarily in the context it may be operating under in terms of shift, state and local colors set on section and module level. If you include other actions in a behaviour with this one, the will all be evaluated before you will evaluate the behaviour of the tied-to HWC. Has a transparent return value.</p>
<p>System Info</p>	<p>Displays system information, mainly status on connections.</p>
<p>No Action</p>	<p>No action. Will blank a display</p>
<p>Wait 1/10s second</p>	<p>Waits for a number of 1/10 second periods.</p>
<p>Custom Handler</p>	<p>Calls a custom handler (which need to be compiled into the firmware. (Special feature).</p>
<p>Inactivate</p>	<p>Inactivates the panel until pressed again. In inactive mode, no keypresses, turned knobs or pulled handles will result in any action. All displays and button colors will remain active though.</p>
<p>Stop Connect</p>	<p>If pressed, any device which is not yet connected will be disabled (until reboot of the controller).</p>
<p>Range Limiter</p>	

Value Scaler	
Panel Brightness <div> System: Panel Brightness 8 </div>	Sets panel brightness to the selected value. May only work for some types of interface components (some buttons). Panel is normally on full brightness.
Local Shift Register <div> System: Local SLR <div> Reg A Reg B Reg C Reg D </div> </div>	<p>Defines which shift register is used in the context to evaluate shift level. The context can be a module/controller, a section and even a single behaviour. In case a local shift register is assigned on a regular HWC, it <i>has</i> to be the first action in the action list, otherwise the behavior will most likely be unintentional.</p> <p>This action does not depend on any trigger from the HWC, it will always be evaluated if inside the proper state and shift levels.</p> <p>Has a transparent return value.</p>
Local State Register <div> System: Local State <div> Reg P Reg Q Reg R Reg S </div> </div>	<p>Defines which state register is used in the context to evaluate controller state. The context can be a module/controller or a section but it will not have any effect on regular HWCs.</p> <p>This action does not depend on any trigger from the HWC (obviously, since it's non functional on a regular HWC), it will always be evaluated if inside the proper state and shift levels.</p> <p>Has a transparent return value.</p>
Local Color <div> System: Local Color Pink <div> Off White Cream Red (B) Rose Pink Purple Amber (B) Yellow (B) Dark Blue Blue Ice Cyan Spring (B) Green (B) Mint </div> Off is dimmed and System: Shift Level Toggle </div>	<p>Sets the color for a button to something else than the default color schemes (default is yellow for bi-color, and white for RGB buttons). It affects SmartSwitches, Bi-color and RGB color buttons only. It also overrides special colors like red and green which are often returned for actions like setting sources on Program or Preview/Preset or recording. A local color action can be included anywhere among the actions executed for a HWC - it doesn't have to be the first action for instance. However, like any other action it is evaluated with respect to states and shift levels. Local color actions can be set also on section and controller level HWCs.</p> <p>The first parameter (in the example "Pink") will set the default color for both on and off (dimmed) state. If the second parameter (in the example "Amber") is set, this color will be used in the off-state of the button. The third parameter determines if the off-state of the new default color appears dimmed (default) or at full brightness. The color called "Default" will reset the color back to the default color schemes.</p> <p>All colors are designed to be distinctly different from each other on RGB buttons, but for bi-color buttons this is only guaranteed for those marked "(B)".</p> <p>For SmartSwitches the Local Color action will affect the smartswitch only if it's included as an action for the SmartSwitch HWC - not on section or module level. Furthermore, Default and Off settings may currently render unexpected results. The use of On or Off color is evaluated based on the binary return value of previous actions in the behavior.</p> <p>This action does not depend on any trigger from the HWC, it will always be evaluated if inside the proper state and shift levels.</p> <p>Has a transparent return value.</p>

<h3>Local Graphic</h3> <p>System: Local Graphic 1</p> <h4>Images</h4> <div> <div>1:  Change Image Delete</div> <div>2:  Change Image Delete</div> <div>3:  Change Image Delete</div> <div>4:  Change Image Delete</div> </div> <div> Save Settings Add Image </div> 	<p>Selects a graphic number to use for this button (in case it has a display). Graphics are uploaded as media on cores.skaarhoj.com for your controller. Files must be 64x32 pixels black and white.</p> <p>This action will always set the graphic no matter if a graphic has already been generated by a previous action in the behavior. The consequence is that you can include Local Graphic before or after other actions and it will overrule their output (normally, only the first action will generate graphic content just like feedback color is generated by the first action that doesn't have a transparent return value).</p> <p>This action does not depend on any trigger from the HWC, it will always be evaluated if inside the proper state and shift levels.</p> <p>Has a transparent return value.</p>
<h3>Local Label</h3> <p>System: Local Label 1 Is Status Clear</p> <h4>Strings</h4> <div> Add String </div> <div> <div>String <input type="text" value=" Line2"/></div> <div>String <input type="text" value="Header Txt Line2"/></div> <div>String <input type="text" value="Header Txt Textline 2 Textline 3"/></div> <div>String <input type="text" value=" Textline 2 Textline 3"/></div> </div> 	<p>Selects a string number to use for this button as label (in case it has a display). Strings are entered as media on cores.skaarhoj.com for your controller. The strings for a controller can be used for many purposes (URLs, options, commands etc). If you use a string as a label, please format it according to "[Header] [Line 1] [Line 2]". You can omit header and line 2 if you want. Whitespace is respected, so you may want to exclude space from around the vertical lines. If two lines are shown, they can be up to 10 characters long (header too), but if a single line is shown, its 5 characters long.</p> <p>If "Is Status" is selected, the button label will be rendered with a solid title bar. This conceptually indicates that the label shows the current status of a value instead of merely what will happen if a button is pushed.</p> <p>This action will always set the graphic no matter if a graphic has already been generated by a previous action in the behavior. The consequence is that you can include Local Graphic before or after other actions and it will overrule their output (normally, only the first action will generate graphic content just like feedback color is generated by the first action that doesn't have a transparent return value). However, notice the function of the "Clear" flag: If set, it will blank the display before setting labels, otherwise it will just set those labels (header, line1, line2) which are not empty.</p> <p>This action does not depend on any trigger from the HWC, it will always be evaluated if inside the proper state and shift levels.</p> <p>Has a transparent return value.</p>
<h3>IP Request</h3>	

<p>Memory Groups</p> <p>System: Memory Group</p> <p>AA 3 Add</p> <p>Label: 0</p> <p>Set/Add Toggle/Add ✓ Add Remove Clear all</p>	<p><i>Binary triggers:</i> Adds, Removes, Toggles the value to the memory group selected (AA or BB). If Set/Add mode is used, the value will be set by a quick push but added by a hold of 1 second. If Toggle/Add mode is used the value will be toggle as the only value in the memory group by a quick push clearing all others except if held down for 1 sec in which case it will not touch other values in the group. Add and Remove simply adds and removes without touching other values in the group. Hold Down will add the value to the group until the trigger is released where it will be removed again.</p> <p><i>Pulse inputs:</i> Will circle through the values from zero to the value given.</p> <p><i>Binary outputs:</i> On when the selected value is found in the memory group (except for Remove which has it reversed and Clear all that is not active)</p> <p><i>Button colors:</i> Follows binary output</p> <p><i>Display text:</i> For displays and smart switches, the value will be shown as a number. If a Label is selected other than "Label: 0" either a label string or an image (1-10) is used instead. Label strings will only overlay the default part (header, line1, line 2) if they are not empty.</p> <p>The first value set in memory group AA and BB will also be set as the value in Mem A and B</p>
<p>Panel Sleep Timer</p> <p>System: Panel Sleep Time</p> <p>INS CP -</p> <p>✓ Default +</p> <p>Off Now! Test (10s) 10 min 30 min 1 hour 5 hours 24 hours</p>	<p>Binary down trigger: Sets the sleep timer interval of the panel. Default is 4 hours.</p> <p>Pulse inputs: Rotates the value.</p> <p>Displays: For buttons, shows the label, for pulse inputs, shows the current value.</p>
<p>Transform 4-Way behaviour</p> <p>System: Transform 4-way Behaviour</p> <p>✓ Original +</p> <p>Rotate CW Rotate CCW Rotate 180 Flip H Flip V Smart menu</p> <p>Block Up Block Down Block Right Block Left Block L-R Block U-D Pass only Up ✓ Pass only Down Pass only Left Pass only Right</p>	<p>Defines how input from a 4-Way button is transformed before interpretation for a given action. The input from a 4-Way button is essentially an up, left, right, or down press trigger which when interpreted as an encoder for instance will result in pulses when pressed on the left and right edges. However, when inserting this transformation on section or controller level it will transform the four directions for that section and when inserted in a given action it will affect any other actions that comes after itself on the given shift level.</p> <p>A "Rotate CW" transformation means that Up / Left / Down / Right will be mapped over to Left / Down / Right / Up. In other words: whatever happened when you pressed Up before now happens when you press the left side and so forth. The remaining transformations "Rotate CCW", Rotate 180, "Flip H" and "Flip V" work the same way. "Original" behaviour back to the original input.</p> <p>The final item, "Smart menu" has a completely different effect. This should find itself as the first item in a behaviour and changes the 4-Way button into a Smart Menu (it may be preceded by a "Transform 4-Way behaviour" in order to modify which edges of the button triggers the smart menu feature). When you press up and down it will change which of the remaining actions below it will be evaluated. The up/down presses themselves will be filtered out and not reach the evaluated action unless its a hold trigger.</p> <p>The Block and Pass options are filters that will either block or pass presses related to the given directions.</p> <p>This action does not depend on any trigger from the HWC, it will always be evaluated if inside the proper state and shift levels. Has a transparent return value.</p>

Force HWC Type

System: Force HWC Type

✓

Binary

Pulsed

Analog

Speed

Auto Shift Level

System: Auto Shift Level

Match Value: 1

✓

Reg A

Reg B

Reg C

Reg D

Level: 0

Mem AA

Defines how a given input is handed over to the *next* action if possible. For example, you may have an encoder which delivers trigger input to an action that can both work with an absolute input (like an absolute zoom position) but also accept a speed input (like zooming in with a certain speed). Regardless of whether the encoders default input is interpreted the one way or the other, you may like to force it from outside. So assuming in this example that encoders result in absolute position change, you can instead force it to "Speed" which in this case is how the action just below the "Force FWC Type" action receives the input.

Notice it only affects the action just below and it may be subject to availability: Not every hardware component may have an interpretation as another type of component.

Will automatically set the shift level in the selected register A-D to the selected value for "Level:" if the value in the selected memory group (Here "Mem AA") matches the "Match value".

This action does not depend on any trigger from the HWC, it will always be evaluated if inside the proper state and shift levels. Fits well in virtual HWCs.

Has a transparent return value.

System Actions and Virtual HWCs

Virtual HWCs such as Controller, Module and Section elements will receive an actDown trigger the first time the behaviour is ever evaluated. This allows you to place system actions such as setting a flag, state, shift level or similar in a virtual HWC and have it set to a particular value when the controller boots.

Triggering Actions from Binary Inputs

Some of our products such as the ETH-GPI Link have a option to interpret binary inputs. This is used in cases where one would like to have routing control or the like from a device such as the Panasonic AW-RP50 or the AW-RP120 PTZ controller.

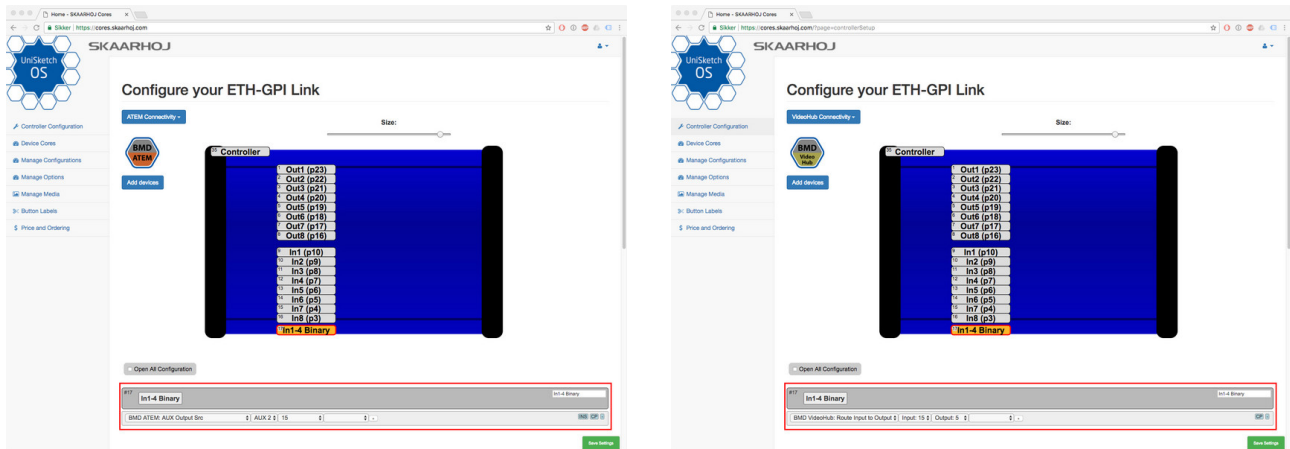
<Output states of GPI OUT1 to GPI OUT4 when the camera number registered to CAM OUT is selected>

	Output of GPI OUT1 to GPI OUT4			
	GPI OUT1	GPI OUT2	GPI OUT3	GPI OUT4
CAM OUT1	On	Off	Off	Off
CAM OUT2	Off	On	Off	Off
CAM OUT3	On	On	Off	Off
CAM OUT4	Off	Off	On	Off
CAM OUT5	On	Off	On	Off
CAM OUT6	Off	On	On	Off
CAM OUT7	On	On	On	Off
CAM OUT8	Off	Off	Off	On
CAM OUT9	On	Off	Off	On
CAM OUT10	Off	On	Off	On

Example of Tally output from the manual of the Panasonic AW-RP120

Fixed Routing

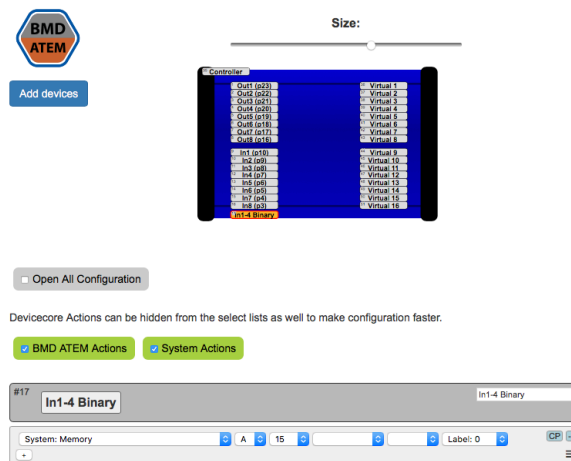
The two below examples shows how a configuration should be made to trigger ATEM AUX routing our Blackmagic Design VideoHub routing. In these cases the routing is fixed so CAM1 from the AW-RPx will *always* correspond to CAM1 in the ATEM and Input1 in the VideoHub.



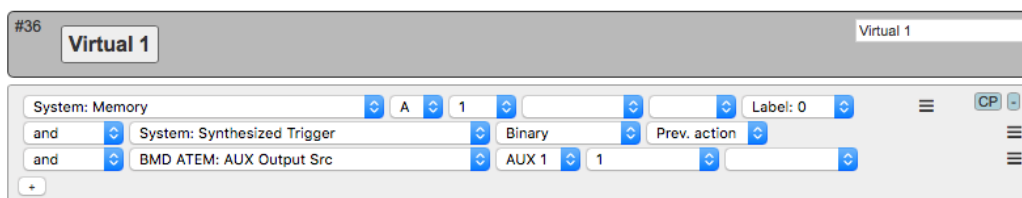
Flexible Routing

In some cases it is not desired to have the fixed coloration between CAM1 on the AW-RPx and CAM1/ Input1 on the ATEM/VideoHub. The guide below explains how you achieve your desired routing.

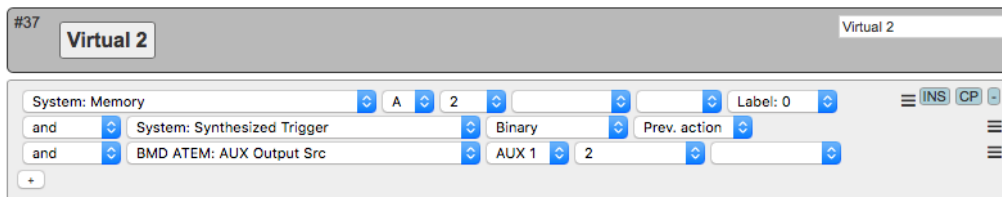
Set the hardware component "In1-4 Binary" to the value below. This will generate a memory parameter, A, in UniSketch which we will use to do our routing. If CAM1 is selected on the AW-RPx the memory parameter will be A=1, if CAM2 is selected it is A=2 and so forth.



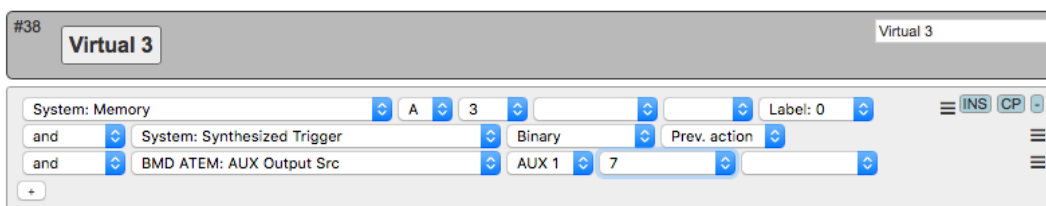
Now we will utilise the Virtual Hardware Components called "Virtual 1" and use the command "System: Synthesized Trigger" to trigger a action. In this case we set it to source 1 for AUX 1 on the ATEM. But you could as well do it for a different source. Notice A=1.



Then you continue by adding more Virtual Hardware Components actions as below. Notice in this case A=2 and we trigger source 2 for AUX 1 on the ATEM.



Now we have A=3 and in this case we trigger source 7 for AUX1 on the ATEM.



And then you continue to achieve the routing you would like. You do not need to different Virtual Hardware Components - more commands can be assigned to just one like illustrated below

