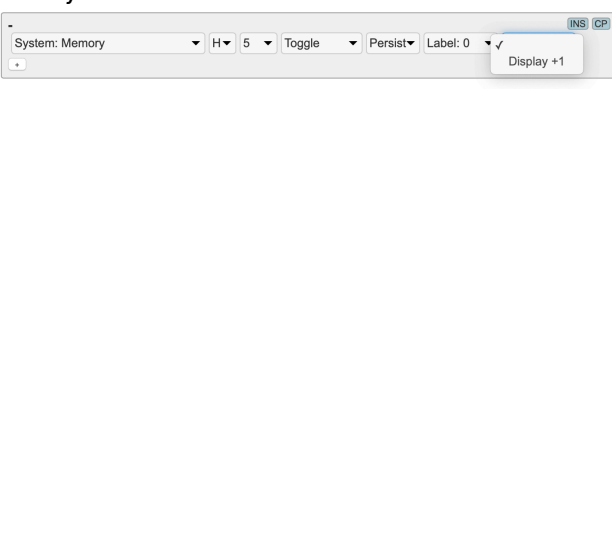
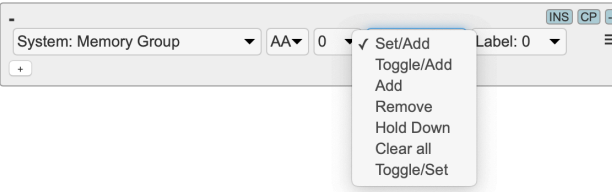
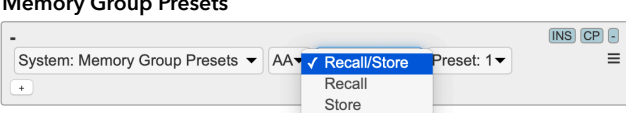
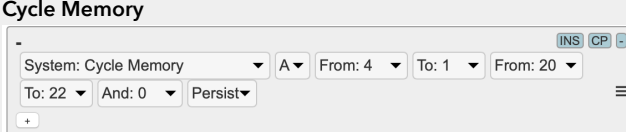



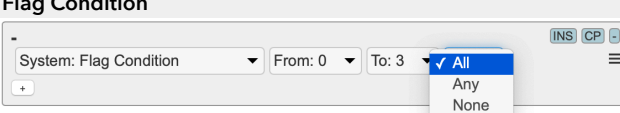
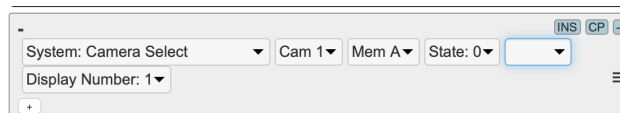
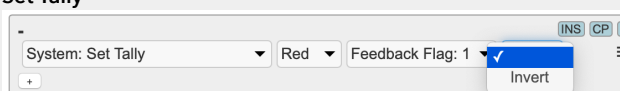
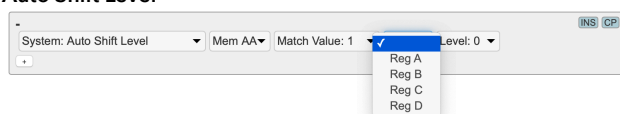
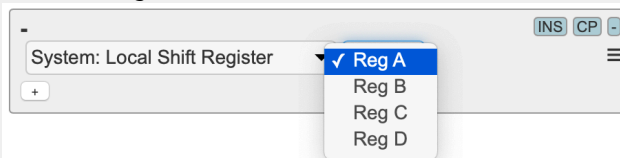
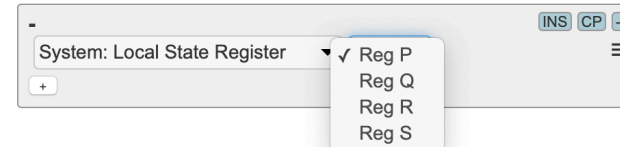
System Actions

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

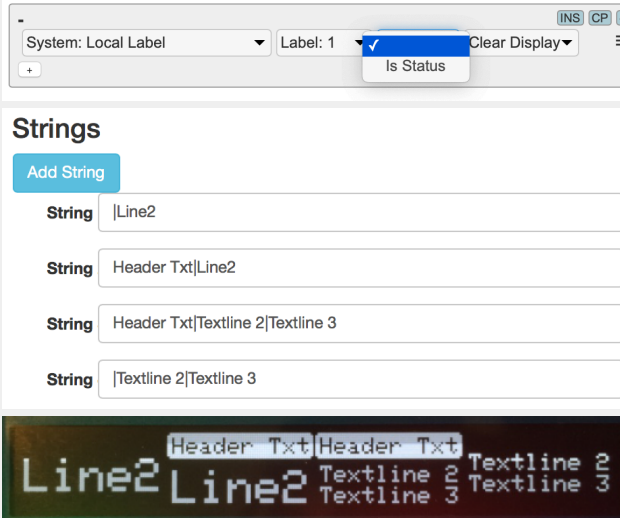
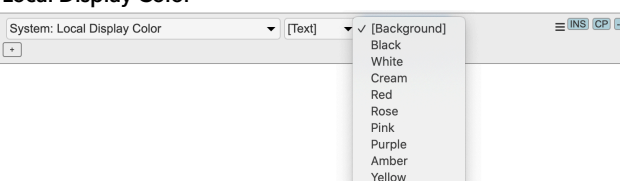

	<p>Sets the controller shift level.</p> <p><i>Binary triggers:</i> Sets the selected shift level for the given register (default or A-D). If Hold Down is selected, the shift level will fall back to Level 0 whenever the trigger is released. Toggle will set the shift level unless it's the same in which 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. If Hold Down - Prev (Previous) is selected, the shift level will fall back to the previous shift level (and not Level 0). If Toggle - Prev (Previous) is selected, the shift level will toggle between the current shift level and the selected shift level on the action (and not fall back to Level 0)</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>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. If Hold Down - Prev (Previous) is selected, the state level will fall back to the previous state level (and not Level 0). If Toggle - Prev (Previous) is selected, the state level will toggle between the current state level and the selected state level on the action (and not fall back to Level 0)</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>

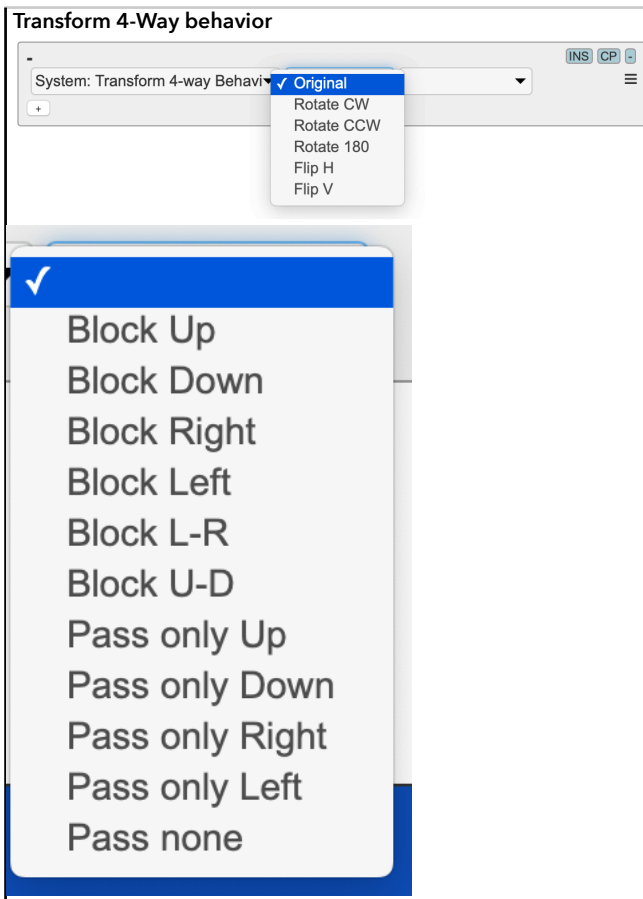
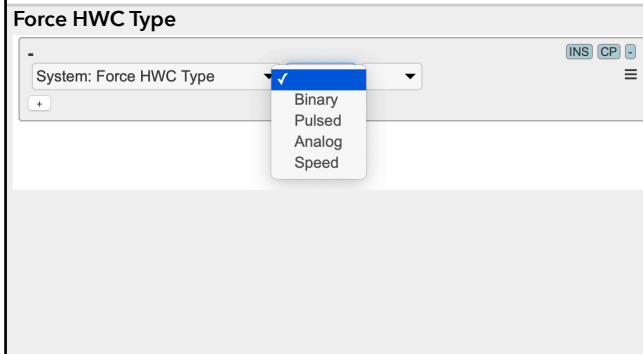
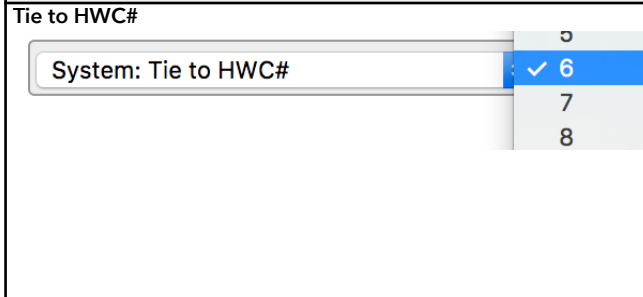
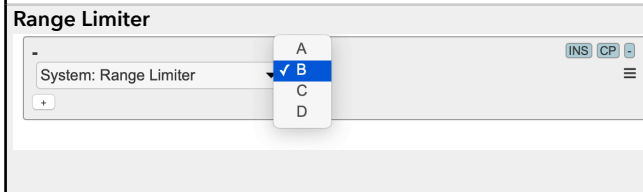
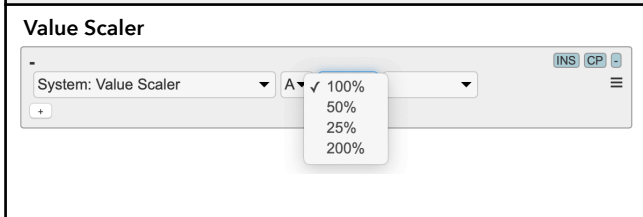
<p>Memory</p> 	<p>Sets memory registers A-N. If "Persist" is selected, the value will be stored in EEPROM and recalled from EEPROM upon booting the controller again (only register A-D). Values set in memory group A and B will also be set in Memory Group A and B.</p> <p><i>Binary triggers:</i> Sets the selected value for the given register A-N. 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), or Grid 1,1-10,1 is used instead. Label strings will only overlay the default part (header, line1, line 2) if they are not empty. If Display +1 is selected the selected memory value will display the value +1. This is useful if referring to for instance a ME selection where ME1 is selected via the value 0. Notice that a custom label may be added as a prefix if the action outputs a number. If this is not desired, please use a "Local Label" with the "Clear" flag instead.</p>
<p>Memory Group</p> 	<p>Control values for Memory Group AA and BB</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. For Toggle/Set when a button is pressed repeatedly, the value is added/removed from the memory group. If the key is held down for a second, the memory group is cleared and only that value is set.</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. Notice that a custom label may be added as a prefix if the action outputs a number. If this is not desired, please use a "Local Label" with the "Clear" flag instead.</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>


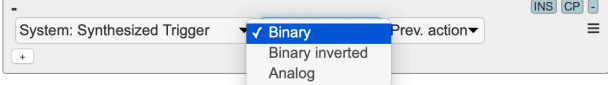
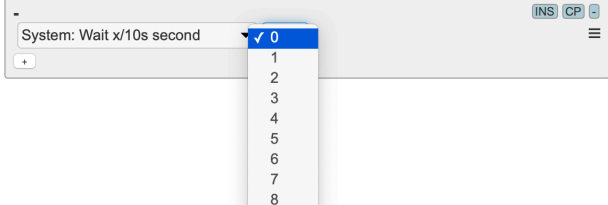
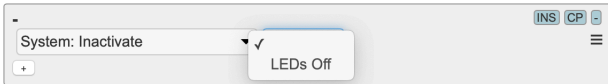
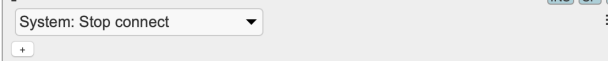

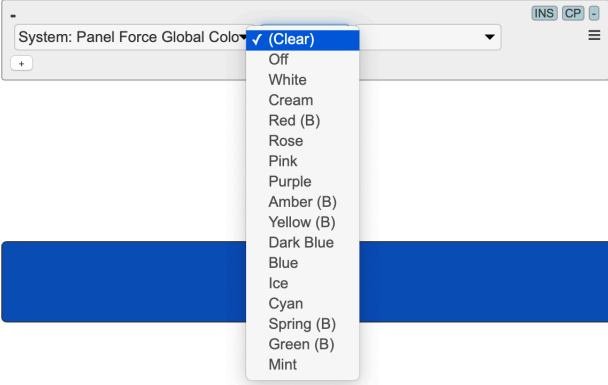
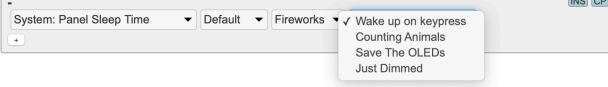
<h3>Memory Group Presets</h3> 	<p>Save, Recall or Save/Recall Memory Group selection for a total of 24 Memory Group Presets per Preset</p> <p><i>Binary triggers:</i> If Store mode, the given Memory Group settings will be saved to the chosen bank. In Recall mode the Memory Group settings will be recalled. If Recall/Store mode the two functions are combined. Press and hold will Store. One press will recall.</p> <p><i>Pulse inputs:</i> Not implemented but Recall and Store will work with long push on encoder.</p> <p><i>Binary outputs:</i> Not implemented.</p> <p><i>Button colors:</i> Highlighted when on.</p> <p><i>Display text:</i> For Store "MGA Prs Set", for Recall "MGA Prs Recall" and for Recall/Store "MGA Preset"</p>
<h3>Cycle Memory</h3> 	<p>Sets memory registers A-N 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>
<h3>Flag</h3> 	<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, line 1, line 2) if they are not empty.</p>

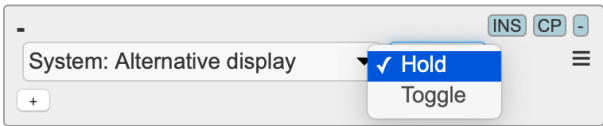
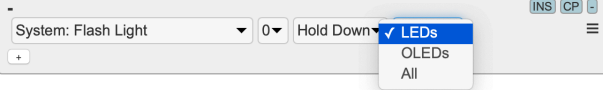

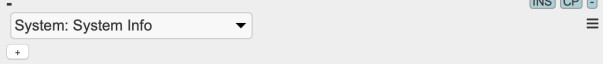
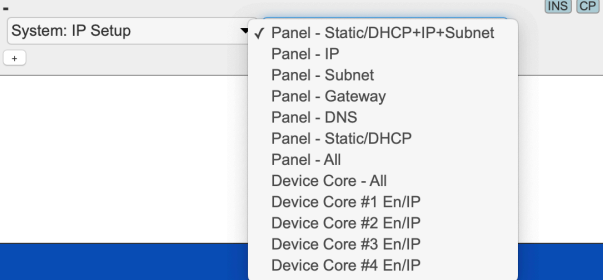
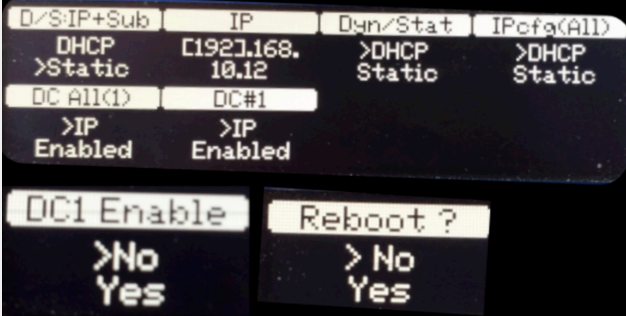
<h3>Flag Condition</h3> 	<p>Output only action.</p> <p>Will output binary true if the range of system flags between from and to (both included) satisfies the condition set in the third drop down. The options are:</p> <ul style="list-style-type: none"> - All: Will return true if all the flags are set, 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>
<h3>Camera Select</h3> 	<p>Combined action setting a Memory value to the chosen memory register (A-D), selecting State (with option for local state register) and select the displayed Camera number. Ideal for configurations with controlling different camera types in different states.</p> <p><i>Binary triggers:</i> Sets memory register, state and display number (1-8)</p> <p><i>Pulse inputs:</i> Not implemented.</p> <p><i>Analog inputs:</i> Not implemented.</p> <p><i>Binary outputs:</i> Not implemented.</p> <p><i>Button colors:</i> Highlighted when on.</p> <p><i>Displays:</i> Shows "Camera/Camera X"</p>
<h3>Set Tally</h3> 	<p>This will color a button, tally bar or the like based if selected flag register is set to true or false. The conditions for true/false can be inverted. Color can be red or green.</p>
<h3>Auto Shift Level</h3> 	<p>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") matchest the "Match 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. Fits well in virtual HWCs.</p> <p>Has a transparent return value.</p>
<h3>Local Shift Register</h3> 	<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>
<h3>Local State Register</h3> 	<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>

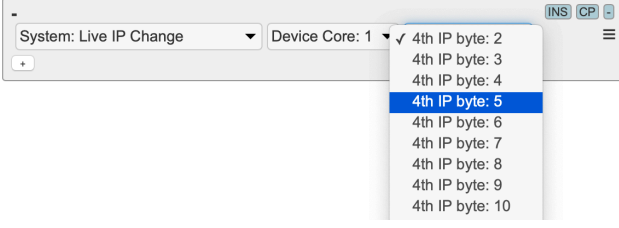
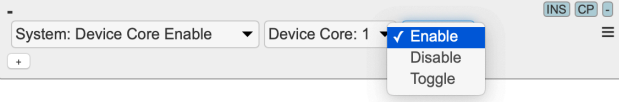
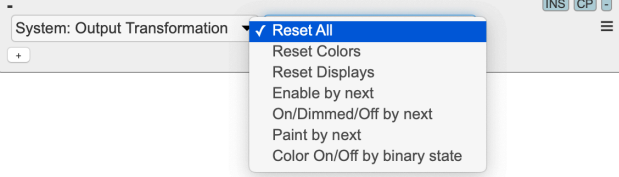
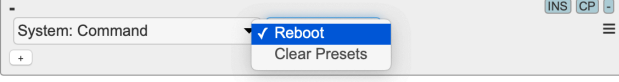
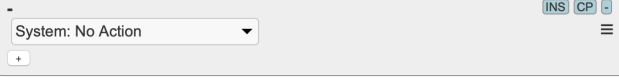

<h3>Local Color</h3>	<p>Sets the color for a button to something else than the default color schemes (default is yellow for bi-color, and white for RGB buttons). It affects SmartSwitches, Bi-color and RGB color buttons only. It also overrides special colors like red and green which are often returned 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 behaviour.</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>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 behaviour. 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>

<p>Local Label</p> 	<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 or in the label grid on the Local web interface (normally used for camera preset labels). 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. If you prefix a label with "\$f2 " the remaining string will be formatted and wrapped as two lines of 5 large characters. If you prefix a label with "\$f3 " the remaining string will be formatted as one line of 3 very large characters. 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 behaviour. 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>If you add a Local Label action after another action that outputs a number value for display (such as System: Memory), the [Line 1] label will be added as a prefix to the value. If this behaviour or if generally the additive nature of Local Label only being applied if the Header, Line 1 and Line 2 values are not empty is not desired, then please select the "Clear" flag".</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>
<p>Local Display Color</p> 	<p>Sets the color for a color display to something else other than the default color schemes (default is white text on black background). A local display 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 display color actions can be set also on section and controller level HWCs.</p> <p>The first parameter [Text] will set the default color for the test. The second parameter [Background] sets the color for the background.</p> <p>All colors are designed to be distinctly different from each other on color displays.</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> <p>*This action is only compatible with controllers with color displays.</p>
<p>Local Display Font</p> 	<p>Allows you to customize the display font parameters.</p> <p>Each parameter can be set from 0 -3. The usual font size used is 3, if set to 0 the default value will be used.</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>Transform 4-Way behavior</h3> 	<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 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>
<h3>Force HWC Type</h3> 	<p>Defines how a given input is handed over to the <i>next</i> 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 this is how the action just below the "Force FWC Type" action receives the input.</p> <p>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.</p>
<h3>Tie to HWC#</h3> 	<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, they will all be evaluated before you will evaluate the behaviour of the tied-to HWC. Has a transparent return value.</p>
<h3>Range Limiter</h3> 	<p>Range Limiter is used in combination with some iris joystick handling actions. It sets an upper and lower limit range, so if there is a mismatch between the iris f-stop values on a lens and sending full open/closed commands to a camera, that the full range on an iris joystick will still work.</p>
<h3>Value Scaler</h3> 	<p>Implemented on pulse inputs.</p> <p>Value Scaler is used in combination with some iris joystick handling actions. Defines the resolution for an iris joystick handling option. Example: Iris is set to 100% and the Value Scaler at 50% is enabled. A full open/closed behaviour of the iris joystick will now result in a range between 50% and 100% and thereby increasing the resolution of the joystick.</p> <p>Implemented on binary inputs.</p>

<p>Custom Handler</p> 	<p>Calls a custom handler (which need to be compiled into the firmware. (Special feature).</p>
<p>Synthesized Trigger</p> 	
<p>Wait 1/10s second</p> 	<p>Waits for a number of 1/10 second periods.</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. However if "LEDs Off" are selected button colors will be turned off when enabled.</p>
<p>Stop Connect</p> 	<p>If pressed, any device which is not yet connected will be disabled (until reboot of the controller). Implemented on binary inputs.</p>
<p>Panel Brightness</p> 	<p>Sets the panel brightness between 0 (off) and 8 (max). If "Persists" is selected the brightness settings will be applied after a reboot of the controller. Implemented on binary inputs and pulse inputs.</p>
<p>Panel Force Global Color</p> 	<p>Forces a single LED color over the whole panel regardless of any function specific, local, sectional or global color.</p> <p><i>Binary triggers:</i> Toggles the selected color on/off</p> <p><i>Pulse inputs:</i> Browses through the system colors</p> <p><i>Analog inputs:</i> Not implemented.</p> <p><i>Binary outputs:</i> Not implemented.</p> <p><i>Displays:</i> Shows the color name.</p>
<p>Panel Sleep Time</p> 	<p><i>Binary triggers:</i> Sets the sleep timer interval of the panel. Default is 4 hours. The Just Dimmed is an option for screensaver which will dim down all displays and buttons but keep content and full functionality of panel. It comes back to full brightness when using the panel.</p> <p><i>Pulse inputs:</i> Rotates the value.</p> <p><i>Displays:</i> For buttons, shows the label, for pulse inputs, shows the current value.</p> <p>Ratio between goats and sheeps: 1 goat per 16384ms and 1 sheep per 1024ms so 1:16 ratio</p> <p>Please Note the Panel Sleep Time is enabled by default to prevent display burn-in. Burn-In is not covered by the 1 year hardware warranty.</p>

<p>Alternative Display</p> 	<p>Used on some controllers such as the Air Fly Pro to switch between the content of the displays for the two bus selection rows. When Alternative display is engaged the functions of the lower bus row will be displayed in the displays of the top bus row.</p> <p><i>Binary triggers:</i> Implemented</p> <p><i>Pulse inputs:</i> Not implemented</p>
<p>Flash Light</p> 	<p>A function to turn all displays and/or all buttons on with a single color - so it could be used as a flashlight, or for debugging reasons (max power consumption, display alignment etc)</p>
<p>Web Config</p> 	<p>Enables or disable web config mode on the controller. Action can be put in "controller" Hardware Interface Component in order to always enable web config mode on. Action is similar to writing "webconfig" in the Serial Monitor. See Installation and Operation manual for additional information.</p> <p><i>Binary triggers:</i> Toggles web config on/off</p> <p><i>Pulse inputs:</i> Implemented with confirm. When changing from ON to OFF or vice versa a question mark will appear. The selection is verified by pressing and holding the encoder. If not confirmed it will fallback to its original state.</p>
<p>System Info</p> 	<p>Displays system information, mainly status on connections.</p>
<p>IP Setup</p> 	<p>Allows for IP controller and device core IP configuration from control panel.</p> <p>Panel- Static/DHCP+IP+Subnet: Choose between Static or DHCP, if Static allows you set the IP and Subnet</p> <p>Panel- IP: Allows you to set only the panel IP</p> <p>Panel- Subnet: Allows you to set only the panel Subnet</p> <p>Panel-Gateway: Allows you to set only the panel Gateway</p> <p>Panel-DNS: Allows you to set only the panel DNS</p>
	<p>Panel- Static/DHCP: Allows you to choose between Static or DHCP, if Static allows you to set the IP, Subnet, Gateway, and DNS</p> <p>Panel- All: Allows set IP address for panel and all device cores loaded on the panel. For the panel it also allows you to choose between Static/DHCP and if Static allows you to set IP, Subnet, Gateway, and DNS; For device cores allows you to disable or disable the device core</p> <p>Device Core- All: Allows you to set IP address for all device cores and Enable or Disable all the device cores loaded on the panel</p> <p>Device Core #x EN/IP: Allows you to set IP address on the selected number device core and Enable or Disable the device core</p> <p><i>Pulse inputs:</i> Will allow you to cycle between options. On hold down selects the option or for setting a address, it sets the number</p> <p><i>Display Text:</i> See pictures for examples Solid Line 1 shows what you will be able to control Line 2 + 3 show configurable options or current IP information</p>

<p>Live IP Change</p> 	<p>Will change the 4th byte of a given device cores IP address, and then “restart” the device core. It can be useful for using a single device core to “switch” between multiple devices, but only connecting to a single at a time. Not applicable for all Device Cores.</p> <p><i>Binary Triggers:</i> Sets 4th byte in selected device core to selected integer from 2-254</p> <p><i>Pulse Input:</i> Not implemented</p> <p><i>Displays:</i> Top Line ‘DC#: DC Type</p> <p><i>Button colors:</i> Highlighted when IP address matches, otherwise dimmed</p>
<p>Device Core Enable</p> 	<p>Allows for the enabling and disabling of a device core without a reboot of a controller. Not applicable for all Device Cores.</p> <p><i>Binary Triggers:</i> For Enable the Device Core is enabled. For Disable the Device Core is disabled. With Toggle it will enable/disable depending on current state.</p> <p><i>Pulse Input:</i> Only implemented for Toggle</p> <p><i>Displays:</i> Top Line ‘DC#: DC Type; Second Line ‘Enabled’ when enabled ‘Disabled’ when disabled</p> <p><i>Button colors:</i> Highlighted when enabled, Dimmed when disabled</p>
<p>Output Transformation</p> 	<p>Allows for the basic color and binary outputs of ‘and’ functions to be modified based on the defined criteria.</p> <p>Function descriptions in “Output Transformation” section further below:</p> <p>Reset All: Reset Colors: Reset Displays: Enable by next: On/Dimmed/Off by next: Paint by next: Color On/Off by binary state:</p>
<p>Command</p> 	<p>For Reboot: Reboots the panel</p> <p><i>Binary triggers:</i> Hold down for 3 seconds reboots the panel. Releasing before countdown ends will stop the reboot.</p> <p><i>Pulse inputs:</i> Hold down for 1 second reboots the panel</p> <p><i>Displays:</i> Top line ‘Reset’ Line 2-3 Hold to reset When pressed, counts down to reset When rebooting displays ‘Resetting Now’</p> <p><i>Button colors:</i> Red on hold down, otherwise dimmed</p>
<p>No Action</p> 	<p>No action. Will blank a display</p>
<p>BP</p> 	<p>Sets the controller into Blue Pill Mode</p> <p><i>Binary triggers:</i> Sets controller into Blue Pill Mode</p> <p><i>Pulse inputs:</i> not implemented</p> <p><i>Displays:</i> Top Line ‘ Blue Pill; Second Line ‘Enable’</p> <p><i>Button colors:</i> dimmed</p>

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.

Output Transformation

Output Transformation modifies the basic color and binary output of a behaviour. (Definition: A behaviour is the collective of actions applied to a given hardware component in a given state - so: all actions listed for a HWC in a state column). Normally, the first action that returns something else than zero (off) will define how a button lights up: Whether it's dimmed or highlighted (or a relay is off or on), or even red, green or yellow as is the case for program/preview related actions. When this action is added to the list of actions in a behaviour it can modify this priority:

- **Reset All:** If set, then the output value is simply reset and whatever action comes next that returns something else than zero gets to set the button color. Display contents also gets reset.

Use case:

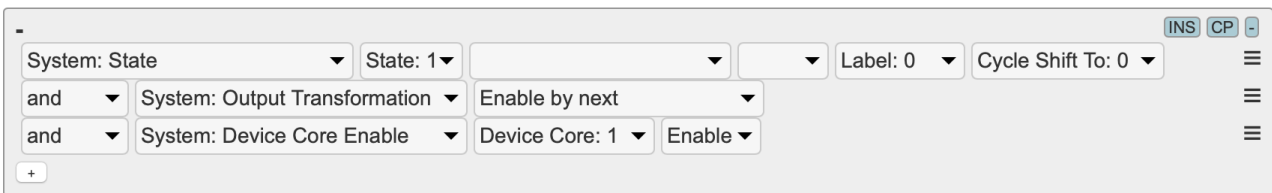
Consider these two actions set for a camera selector. The intention is to select a camera and also send a route to an AUX output on a switcher when the button is pressed. If there is a connection to the camera, the buttons will light up with a dimmed or highlighted color, and the state of the routing is suppressed. However, if the camera device core returns zero (if no camera is detected), then the state of the routing will shine through. You most likely want the camera selector to always dominate in this case, and using an Output Transformation can help by swapping the order of the two actions and insert a reset in between. In that case, it's always driven by the state of the camera selector:

- **Reset Color** resets only the color but passes on display content
- **Reset Displays** resets only displays, but passes on color

- Enable by next:** In such a case, the output value before the output transformation will be preserved only if the action after the output transformation returns true, otherwise it will be reset. The first action after the output transformation will not receive any triggers in this case. Notice that the actions before the transformation will receive triggers - the output transformation only “hides” them based on the criteria of the return value of the action after the output transformation.

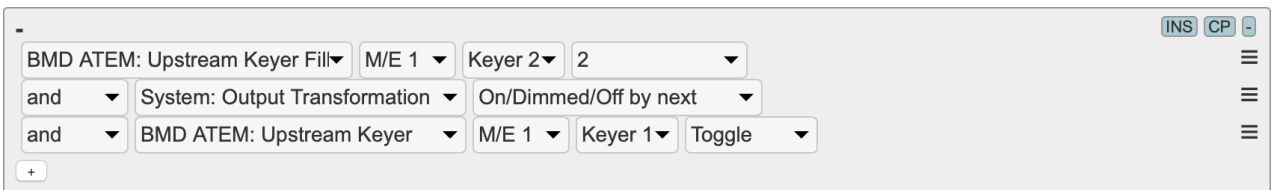
Notice: Some Camera Select actions will not work properly as the action after the output transformation - this is because they manipulate the return values directly which in this case destroys the logic.

Use case could be a state selector which you want to de-color in case a device core is not enabled. This would be done with the action below.



- On/Dimmed/Off by next:** In this case an “AND” operation is performed between color states of the previous and next action. If the first action returns an highlighted color, it will only be shown as highlighted if the action after the output transformation is also highlighted. The first action after the output transformation *will* receive a trigger (unlike how “Enable by next” works).

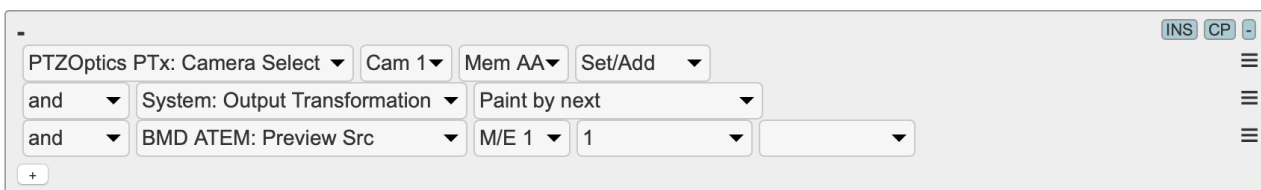
Use case could be a key that selects both a source for the keyer and then toggles the keyer. In that case may want to only show the key as highlighted if the assigned source is selected *and* if the keyer is toggled on. This is achieved like this:



- Paint by next:** In this case, the dimmed or highlighted state that exists before the output transformation will be preserved, but colored with the color coming from the action after the output transformation. The first action after the output transformation will not receive any triggers in this case. The coloring that can get picked up from the action after the output transformation is only yellow, red or green as they come out of ...

Notice: Some Camera Select actions will not work properly as the action after the output transformation - this is because they manipulate the return values directly which in this case destroys the logic. Also, System Actions cannot be used to paint previous actions (because the feature is mainly designed to paint with tally colors from device core actions).

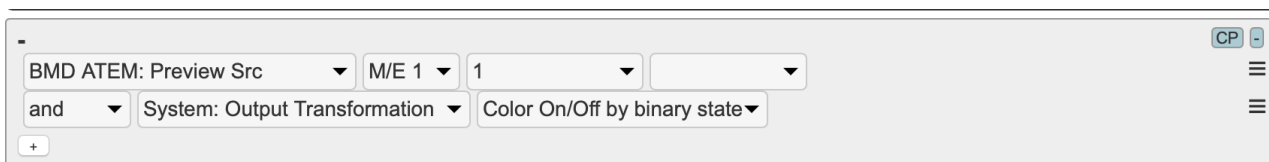
Use case could be a camera selector action where you want color the buttons red/green with tally information from a vision mixer (both in case of highlighted and dimmed):



Additional example provided in the section “Combining Camera Select and Preview/Program Select” further below.

- Color On/Off by binary state:** In this case, the return value that exists when the output transformation is called will be transformed to be either on or off based on what the binary flag of the original return value was.

For example, an action which uses highlighted (binary on) / dimmed (binary off) and possibly colored with special colors like red, green or yellow (such as tally related actions) will not only be highlighted (on) or completely off, and that is with the default color at that hardware component.

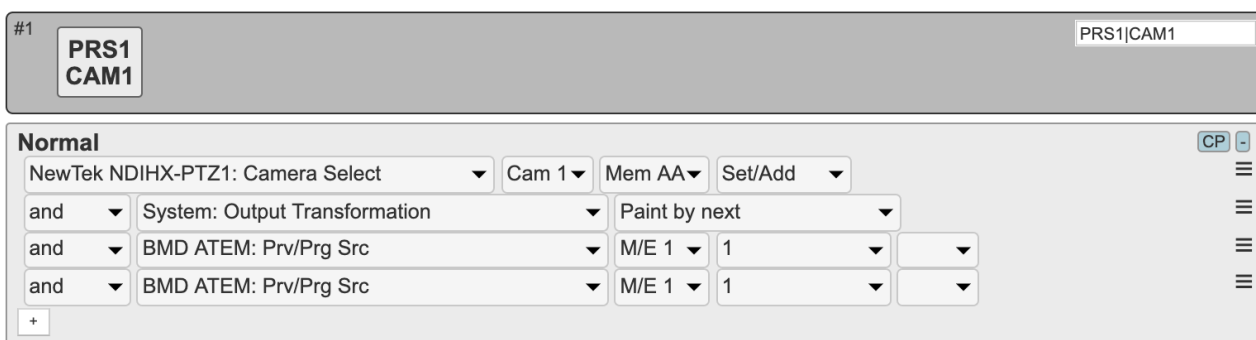


Combining Camera Select and Preview/Program Select

In cases such as on the PTZ Wiz where you have a button functioning as a camera selector and for setting preview/program select on a ATEM Switcher it is desirable to have

- Proper feedback in a button if camera have been selected
- Proper feedback in a button if that particular camera is on preview/program (or if the preview/program source have been changed elsewhere)
- Proper feedback for Camera Select or Preview/Program Select if one of the Device Cores are not connecting

To achieve the above the Output Transformation Paint by Next is *utilized* but the preview/program action **must** be added as an additional action as the Paint by next will not trigger the preview/program action but simply use it to reflect colors. A example is given below



Virtual Triggers

Virtual Triggers are available on all UniSketch OS controllers and allow the user to add increased functionality to a controller where a hardware component might not be needed.

Virtual Triggers (Alpha)

Source states	Operator	Actions
Add trigger		

There are three parts to a Virtual Trigger:

Source state- The action(s) you want to trigger another event

Operator- Sets the conditions for the trigger. Setting to Active enables the trigger. Invert will execute the trigger when the conditions of the Source State are not TRUE. AND (all) will activate this trigger if ALL specified conditions in source state are TRUE. OR (any) will activate trigger if AT LEAST ONE source condition is TRUE. Lastly, XOR will activate trigger if ONE OR THE OTHER (but not both) source state conditions are TRUE. This could be interpreted for multiple sources as "if there is at least one different value, f.ex. 5 sources are TRUE and one is FALSE". If all are TRUE or all are FALSE, this will not activate.

Actions- The action(s) you would like triggered by another event

Example: setting camera tally from a BMD Atem using Virtual Triggers.

