Lorby “Axis And Ohs” StreamDeck Plugin Documentation

Version 73.01 - 27.10.2026

© 2021 Lorby-SI http://www.lorby-si.com

This plugin for the Elgato StreamDeck uses the AxisAndOhs WebAPI.
!Version 3.02 or later of AxisAndOhs is required!

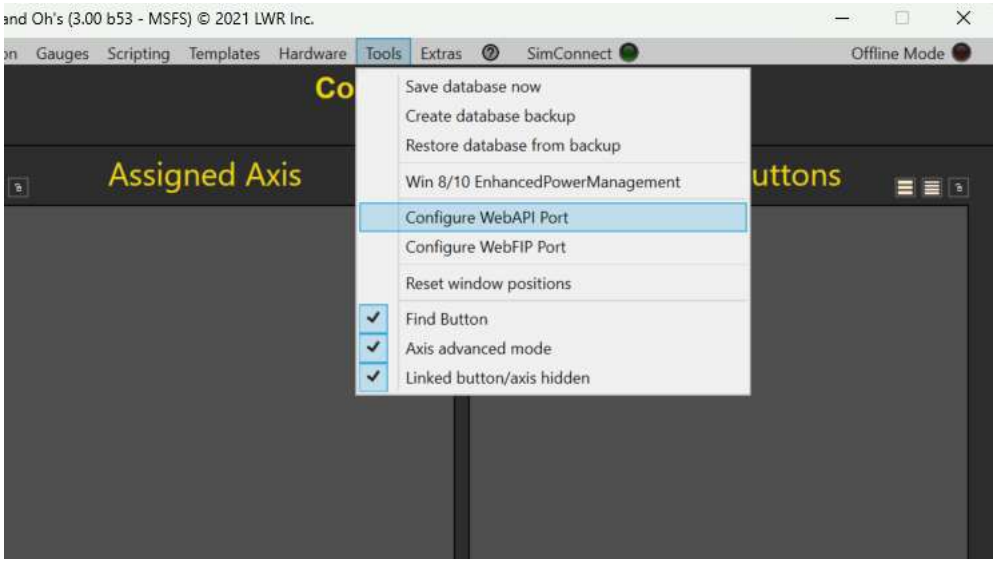
Always start AAO „As Administrator“ from the right-click context menu (or set it to run as administrator by default on Properties->Compatibility). The WebAPI will be enabled automatically – the status is visualized with the two small LEDs in the bottom right corner of the main AAO window.

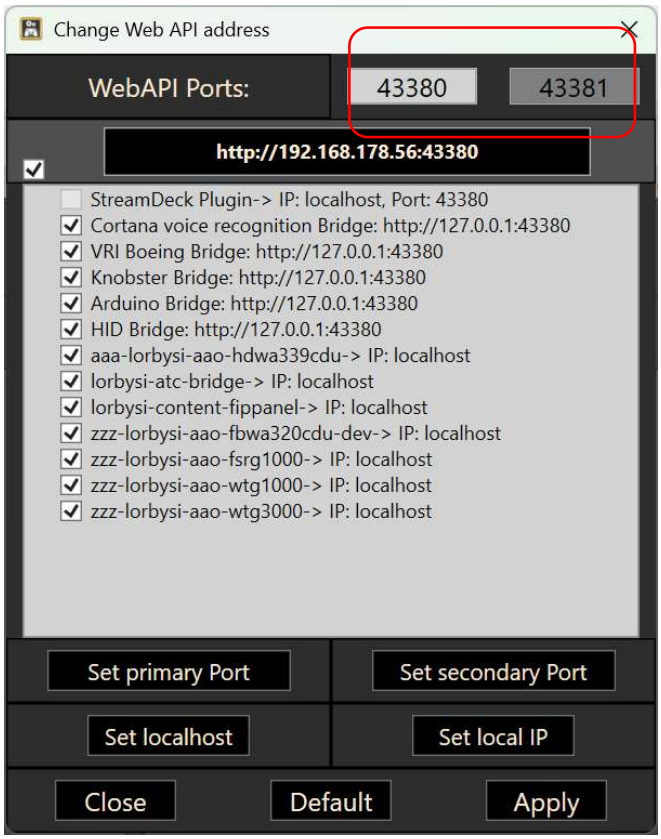
To check if a connection is possible, use the “Connection Status” button action of the plugin – pull it onto an empty spot on the stream deck. Then connect AAO to the simulator and wait until you are sitting in the cockpit. At this point the “Connection Status” button should turn green.

If the connection is blocked by your PC:

If you have another app on your computer that is already using 43380, the port must be changed in the AAO Tools menu.

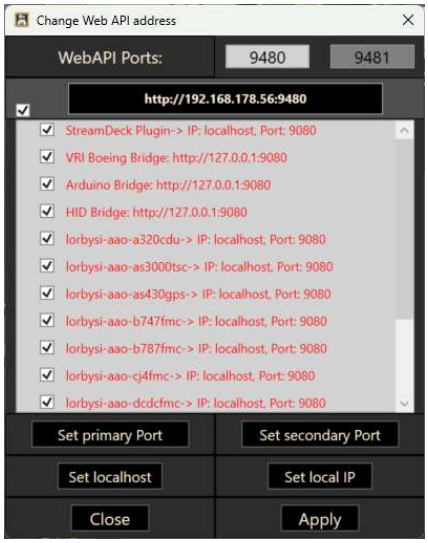
a. The StreamDeck software must be terminated properly before doing this. Right click the StreamDeck symbol in the system tray, there is an option to shut down StreamDeck at the end of the menu
b. In AAO go to „Tools->Configure WebAPI Port“





The WebAPI can be reached on two different ports, the primary (that you can change) and the secondary, which is always “primary + 1”.

After changing the port, you will notice that the dependent AAO addons are listed in red:



The port has to be adjusted in the AAO addons as well. Select the desired addons in the list with the checkboxes on the left, then use “Set primary Port” or “Set secondary Port”.

If you have multiple addons that access the WebAPI, assigning half of them to the secondary Port can help with performance.

You can also exchange “localhost” with the current local IP, should that be necessary on your computer.

Firewall:

You may have to open/allow access to the port in your local firewall by creating incoming and outgoing rules for the TCP port that you have selected above (43380,..).

It is not enough to allow the AxisAndOhs application through the firewall! There must be specific rules for the port.

Plugin Settings :

C:\Users\...\AppData\Roaming\Elgato\StreamDeck\Plugins\com.lorbysi.aao.sdPlugin\settings.js

The file can be edited with a simple text editor:

```
const AAO_URL = "http://localhost:43380/webapi";
const REFRESH_MS = 100;
const REPEAT_MS = 250;
const LONG_CLICK_MS = 750;
```

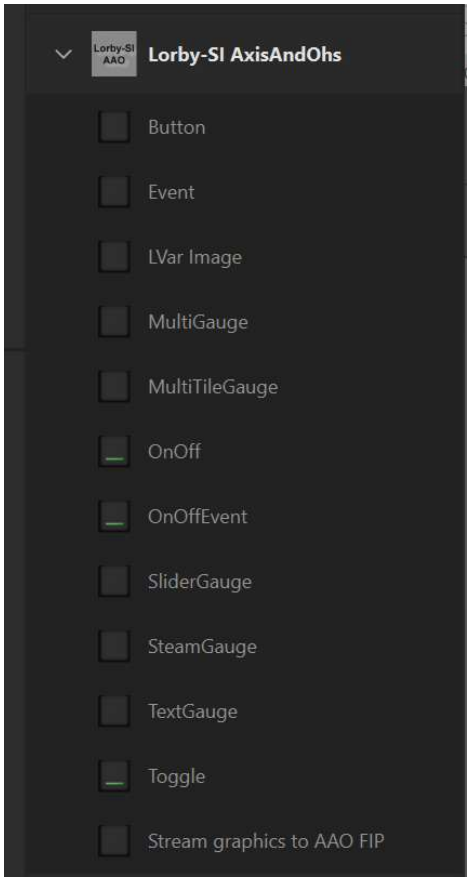
If you are experiencing performance problems with the Elgato app or the StreamDeck, you can slow down the refresh rate of the plugin with the second parameter “REFRESH_MS”.

- A value of 1000 equals one refresh every full second.
- Don’t go below 100 (refresh happens 10 times a second)

You can also change the speed at which a button that is being held down repeats the associated action (250ms = 4 times per second) or when the “long click” option kicks in (750ms by default)

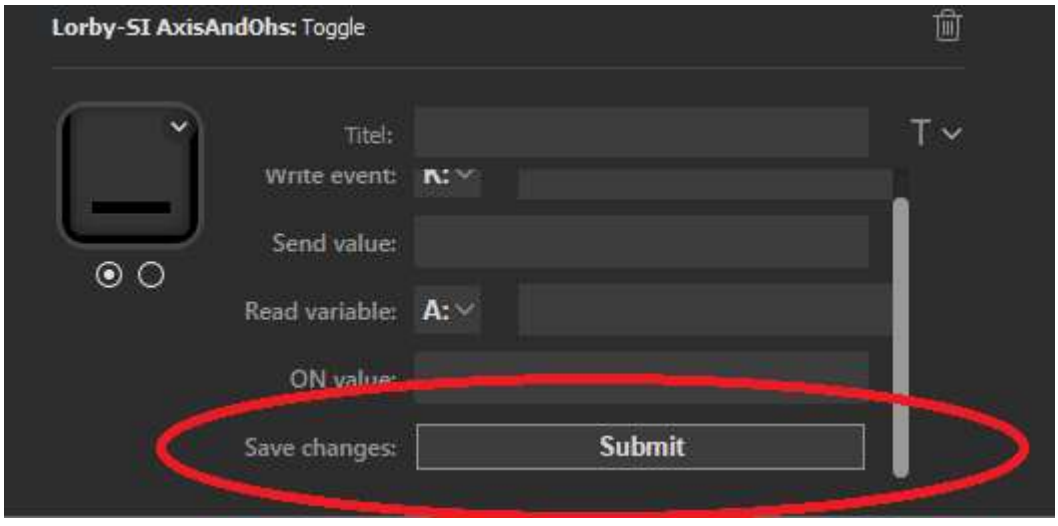
The Elgato software must be shut down and restarted after these changes. Just closing the Elgato Window is not enough, you must right-click on the Streamdeck symbol in your system tray and select the “Quit Stream Deck” option (bottom of the list)

The Plugin provides several different types of buttons for the StreamDeck.



Every button type that you place on the stream deck has properties assigned to it. When you change those properties, make sure to press the "SUBMIT" button at the bottom of the property inspector area in the Elgato software, otherwise your changes will not be applied to the button.

On most actions you can also use "Alt + S" to save.

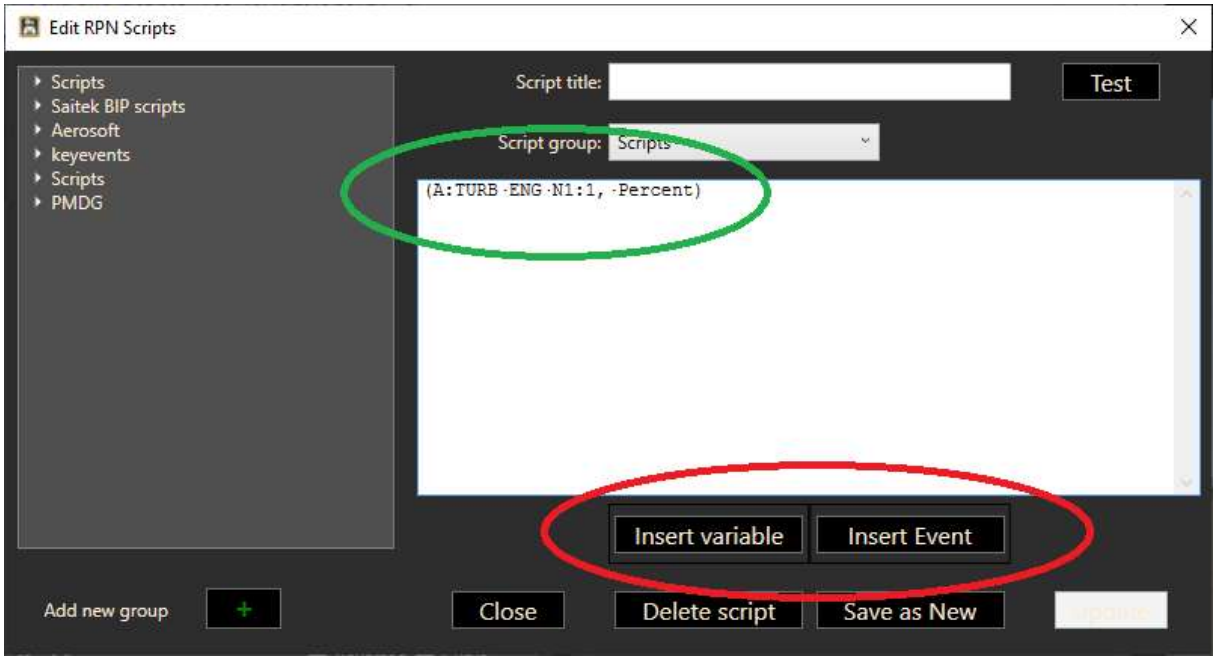


Simulator Events and Variables

This plugin will use the events and variables that are defined for the compatible simulators.

You will need either the SDK documentation for the simulator or make use of the RPN script editor in AxisAndOhs.

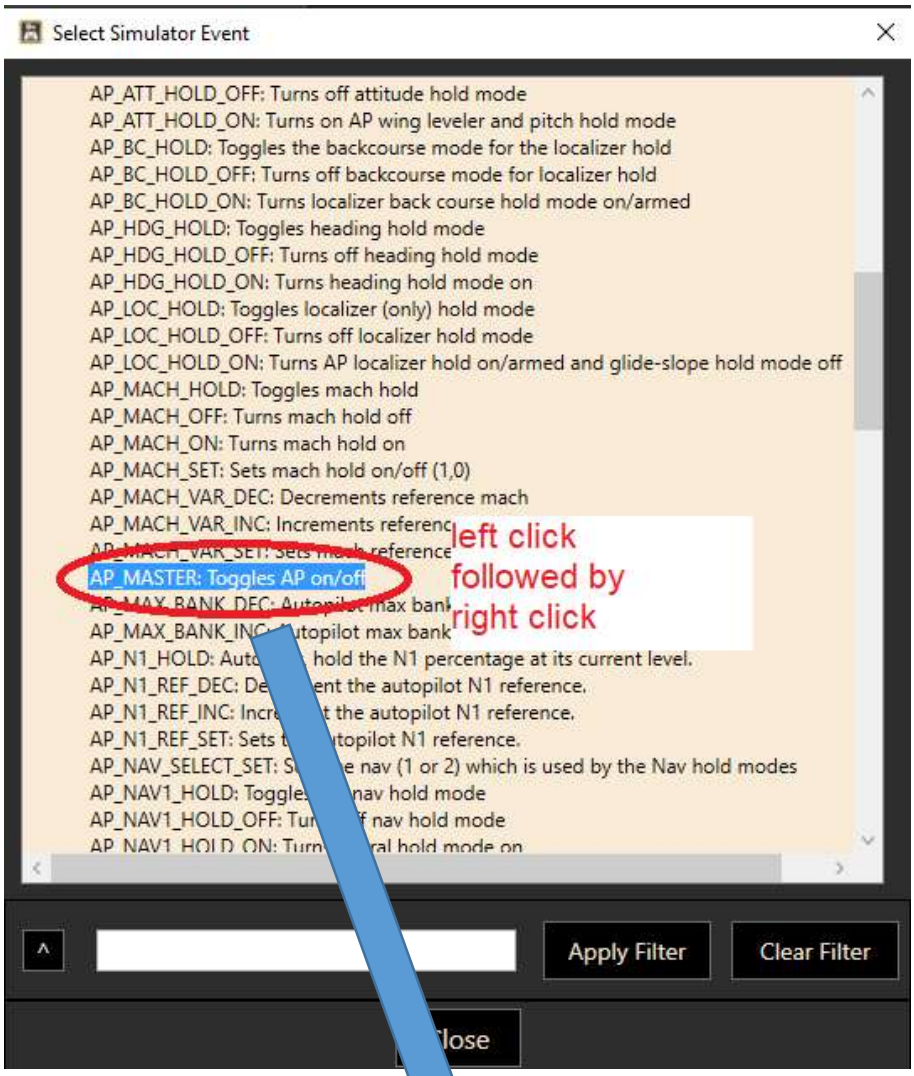
In AxisAndOhs you go to “Scripting -> Manage RPN scripts”. Use “Insert variable” / “Insert event” to find out the exact syntax of the required asset.



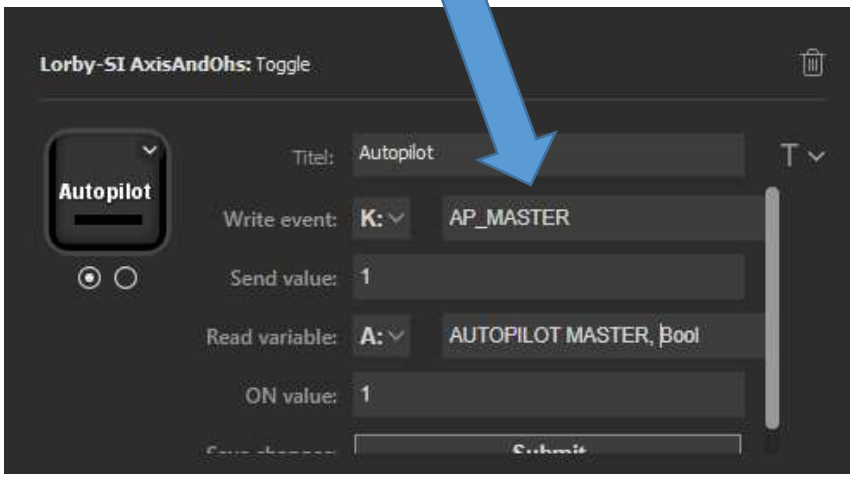
You can also request values that have been calculated by an AAO RPN script, using

- „L: scriptgroup-scriptname“ for numerical values, calculated by a script like this one:
  „(A:INDICATED·ALTITUDE,·Feet)·10000·+“
- „L scriptgroup-scriptname, String“ for string values that are produced by a script like this:
  „%(A:INDICATED·ALTITUDE,·Feet)·10000·+%!5d!

When the AAO variable/event lists are open, you can copy the name of the variable or event to the Windows scratchpad by left clicking then right clicking the item in the treelist. Then you can insert it into the textbox in the Elgato software by pressing Ctrl&V



CTRL & V



“Long Click”

When you hold down the button of any action, it can trigger a “long click” event that can be assigned in the property inspector too. This is a different event from what it does when you just tap the button. You can enter the timeout value that constitutes a long click, the default is 750 milliseconds.

Please note: “Repeat” will only affect the “Long Click” when both options are activated.

“Autoscript”

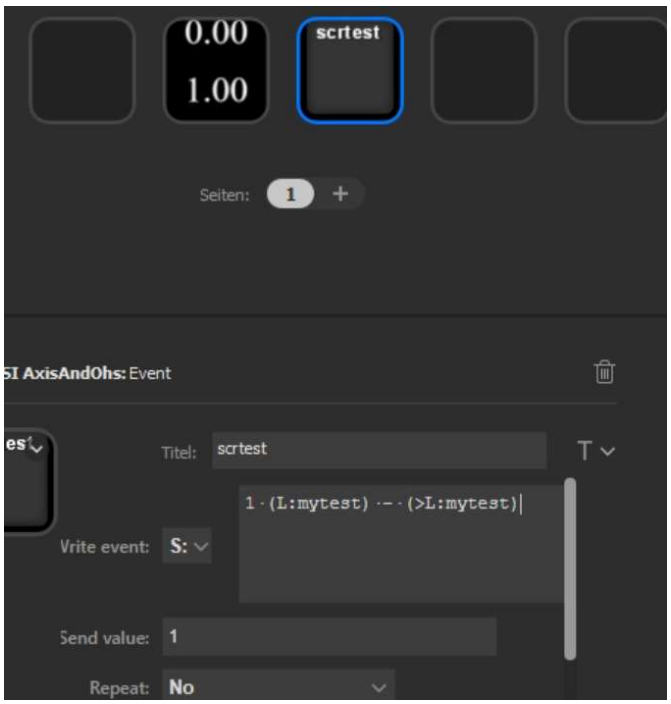
For each action you can call a script repeatedly every 100ms (“Fast” rate) , every 500ms (“Slow” rate) or “One Shot” (script is only executed once)

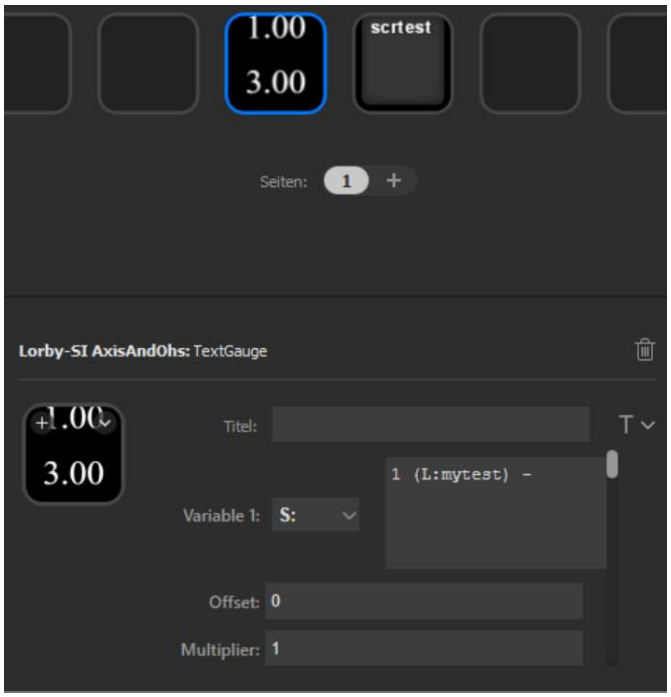
You can either call an AAO script by its ID: “K: groupname-scriptname” or enter RPN code directly (S:)

Utilizing AAO RPN Scripts:

Scripts that are present in AAO can be triggered with K: *scriptgroup-scriptname*

It is also possible to send RPN script code to AAO using the “S:” option with the “Write event” and to read the calculated result of script code by using “S:” on the “Read variable” fields

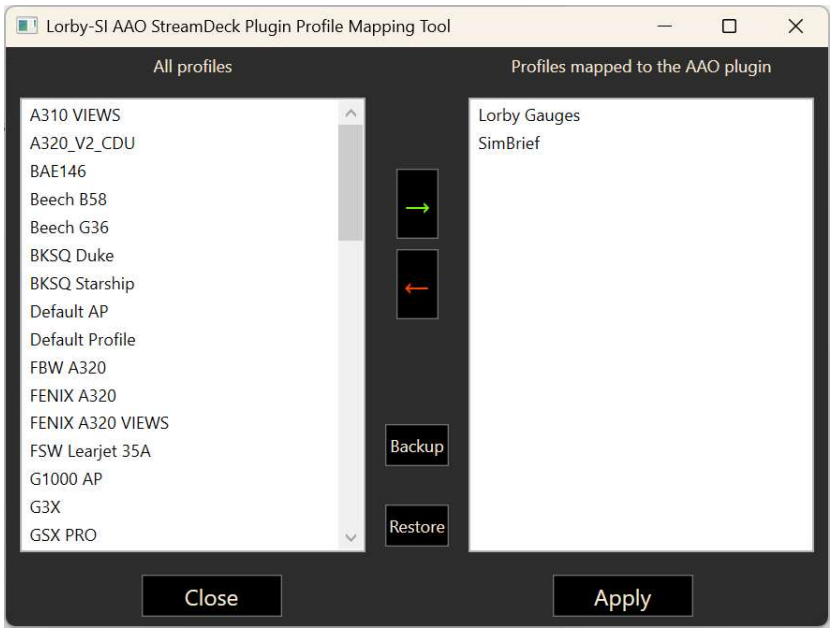




Switching to other profiles using scripts:

The Stream Deck SDK doesn't allow switching to user profiles programmatically. To be able to switch to a different profile with an AAO script this profile must be "bundled" with the plugin itself.

With the "SdProfileMappingTool.exe" you can work around this limitation and map profiles of your choice to the plugin manually. The StreamDeck software must be restarted after making these changes.



Once profiles have been mapped, you can make the Stream Deck call them up using AAO scripts:

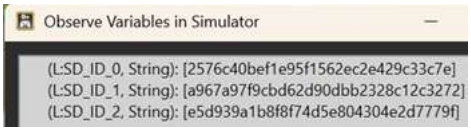
```
(EXEC:streamdeck://plugins/message/com.lorbysi.aao/switchprofile?profname=SimBrief&page=0&streamdeck=hidden)
```

Make sure to include the page number that you want to show (first page = 0)

If you have more than one Stream Deck, you can include the unique device ID in the command. Otherwise, all your decks will try to switch to that profile.

(EXEC:streamdeck://plugins/message/com.lorbysi.aao/switchprofile?deviceid=a967a97f9cbd62d90dbb2328c12c3272&profname=SimBrief&page=0&streamdeck=hidden)

The device IDs can be read from LVars: (L:SD_ID_*n*, String). "n" is the index of the device, starting from 0.



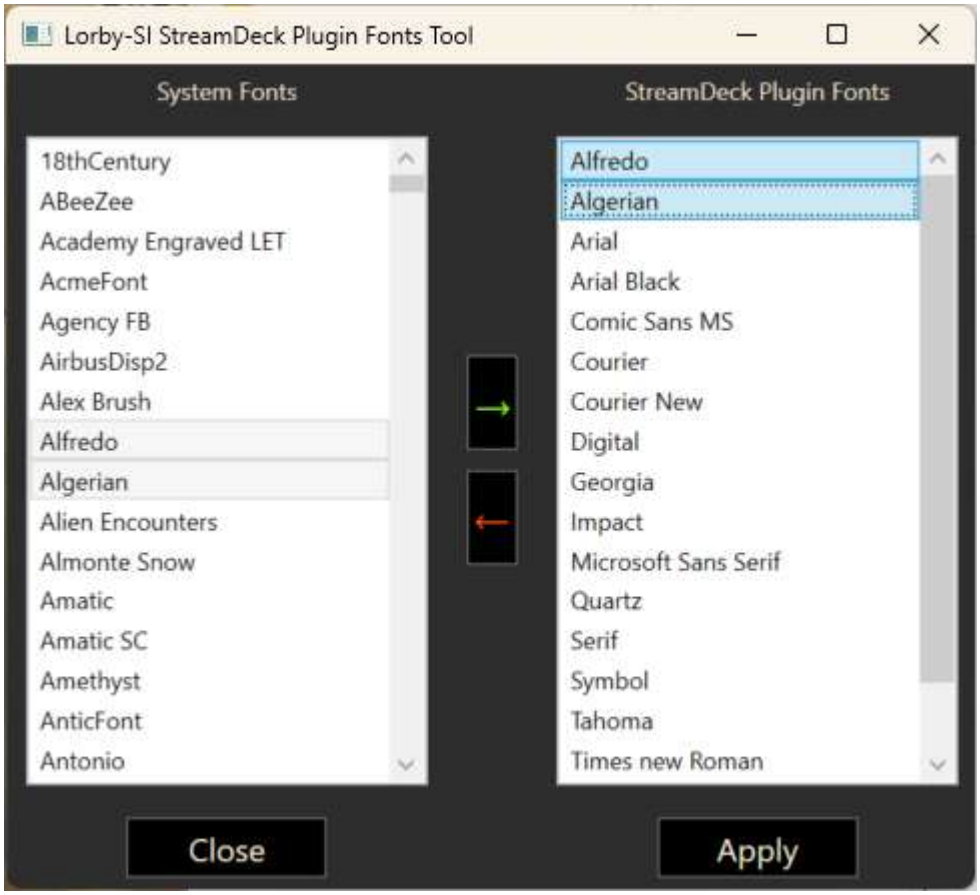
Fonts:

Various Actions can display text elements with different fonts that can be selected from a hard coded list. The fonts on this list should be available on most PCs.

You can add more fonts to this list using the included “SdFontsTool.exe” app.

Be mindful that this will only work on your computer.

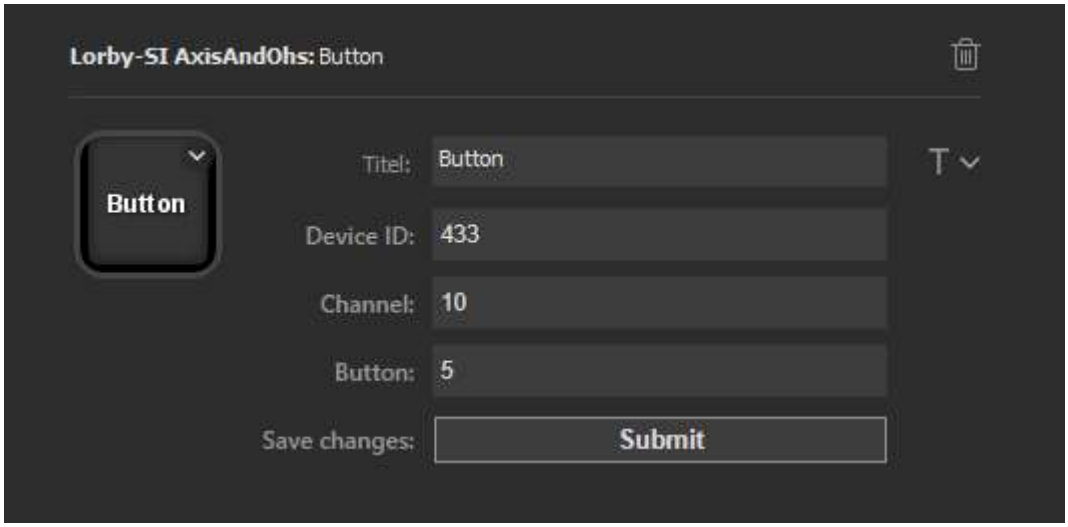
When you are adding exotic fonts, you won’t be able to share your profiles with others. Bear in mind that most font files will be copyright protected, and you are not allowed to just share them with other users!



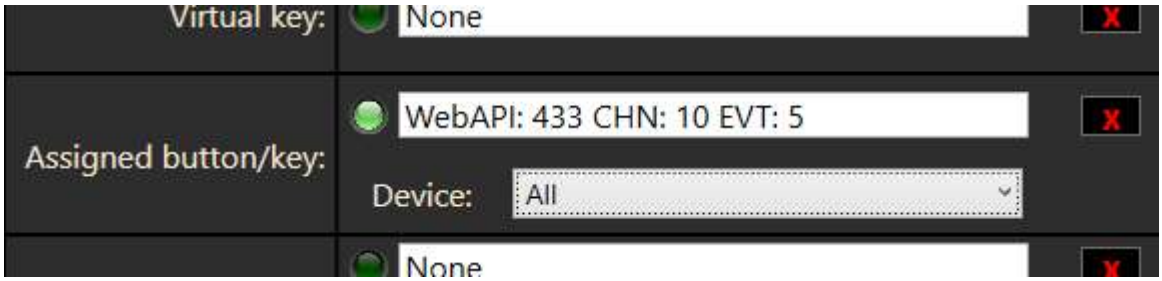
The StreamDeck software must be restarted after making changes to the fonts list, using a right-click on the StreamDeck icon in the system tray.

Type “Button”

This simulates a button press on a Joystick.

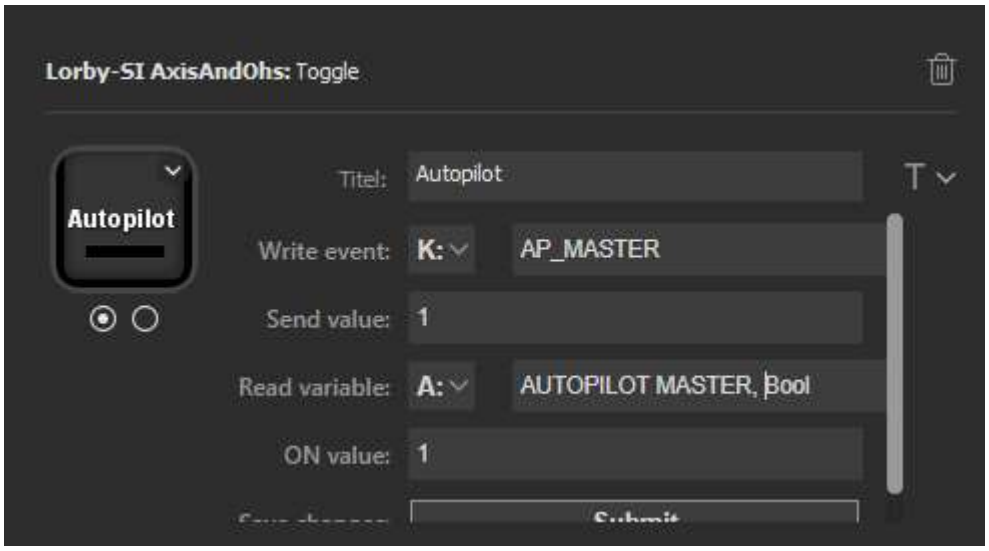


In AxisAndOhs, pressing the button will be registered on the “Add/Change” button dialogs like this:



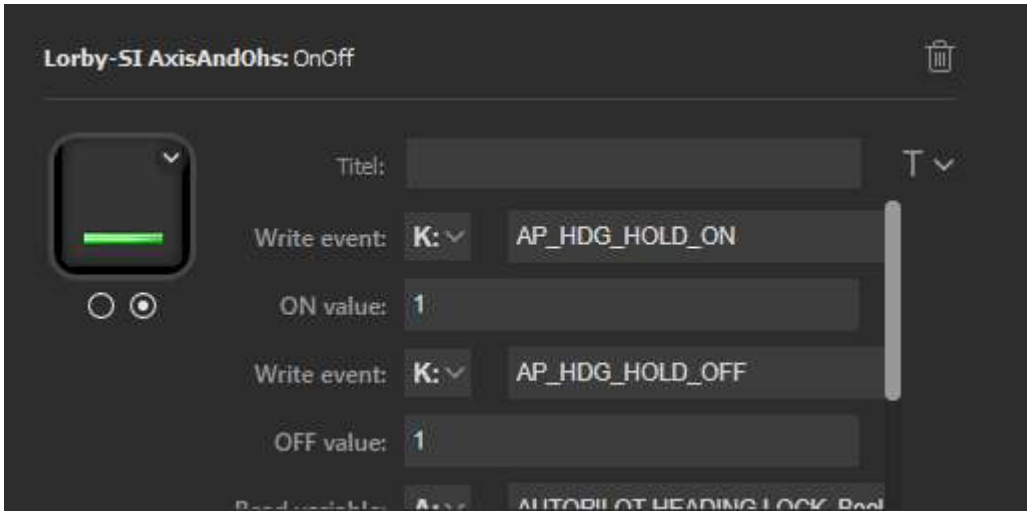
Type “Toggle”

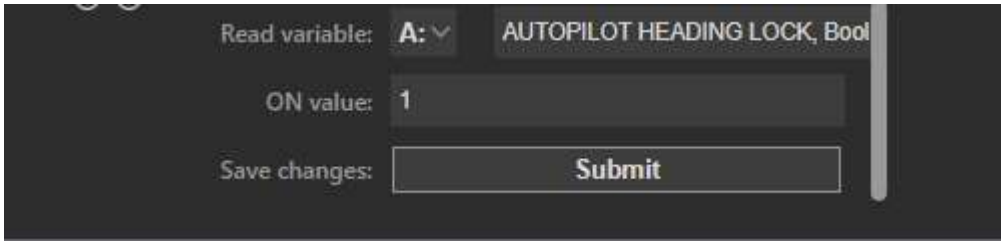
If the desired simulator action is bound to a single toggle event, use this button type to switch between the two states. A simulator variable can be used to receive feedback about the success of the toggle. When the ON value is received, the plugin will load the “on” picture for the button, otherwise the “off” picture is set.



Type “On/OFF”

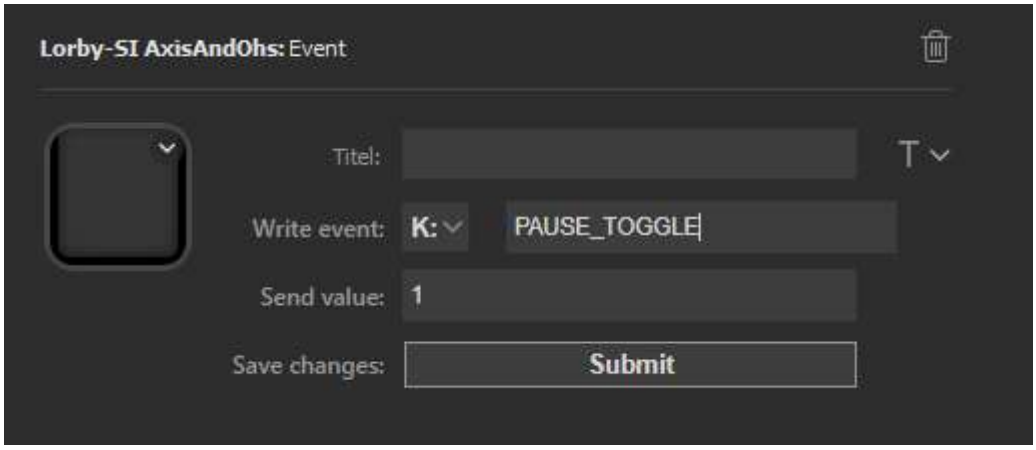
In this case there is no toggle event in the simulator, but separate ON and OFF events instead:





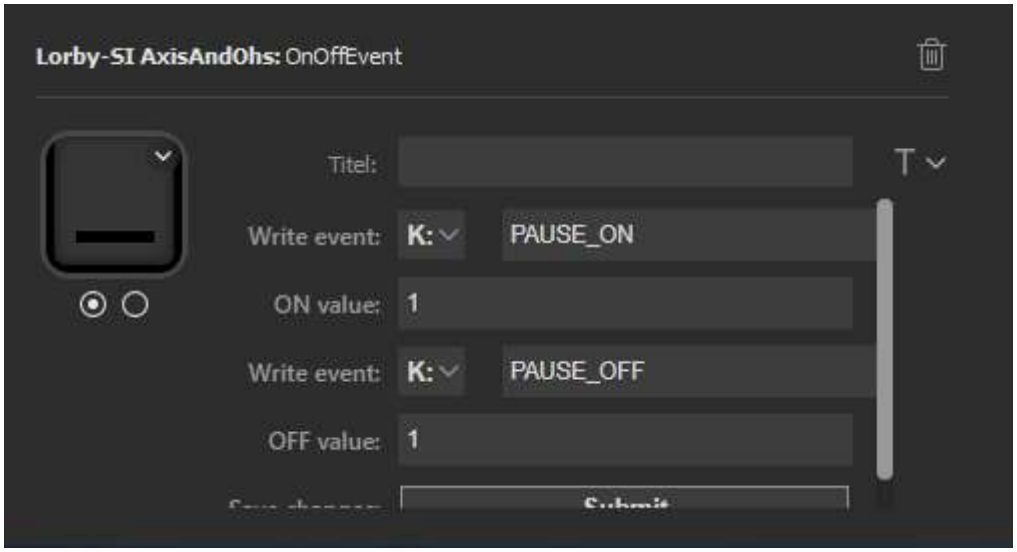
Type “Event”

Send a simulator event directly without querying a variable for a result



Type “DualEvent”

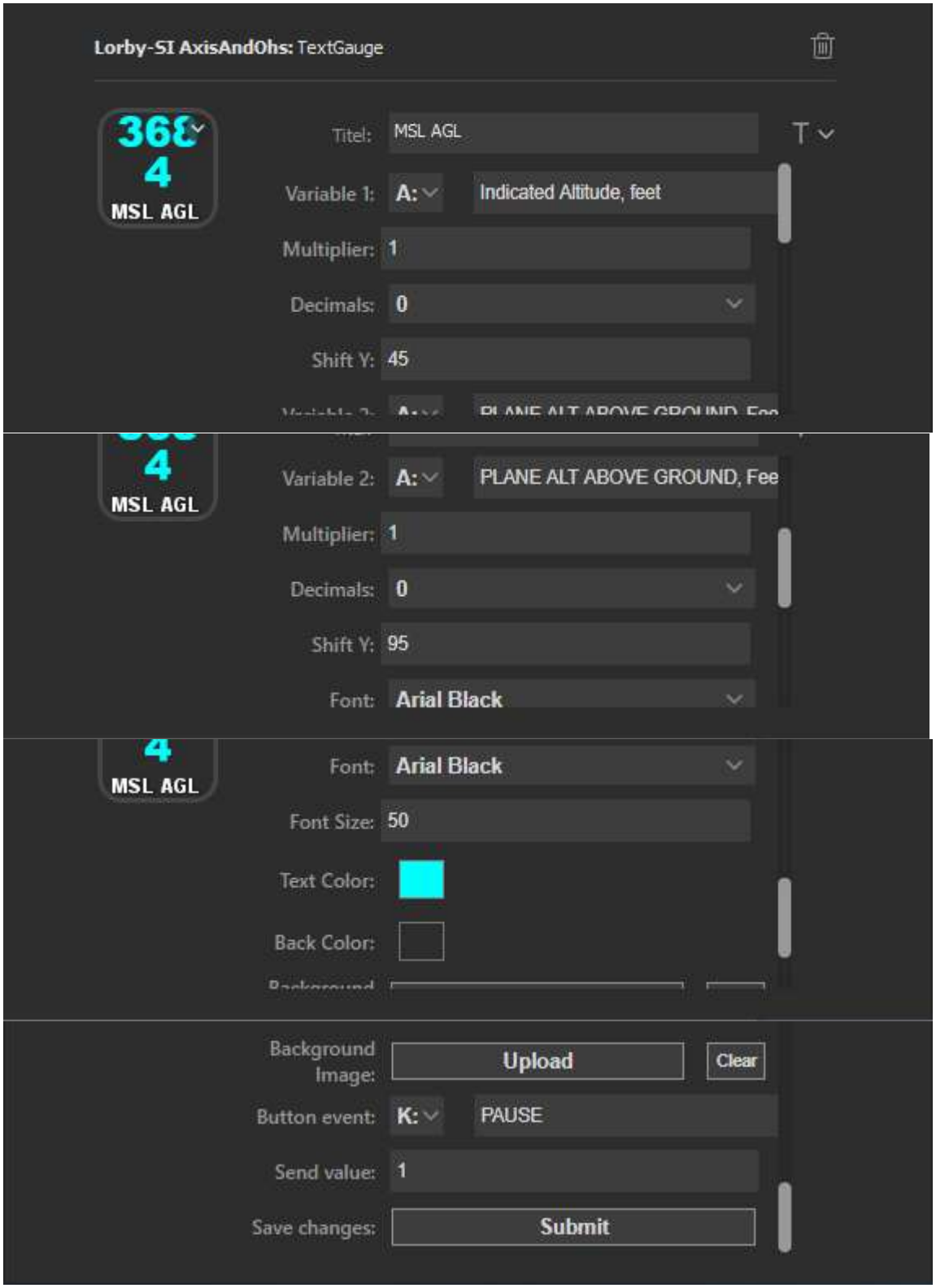
Send separate ON/OFF simulator events directly without querying a variable for a result



Note: all of the following “Gauge” type actions can also be used to trigger events. At the end of the parameter list you will find a “Button event” box for K:, L: and H: events/Variables with the “Send value” right below. When you press the gauge on the StreamDeck, it will send this event/ change this variable.

Type “TextGauge”

This button type will display the values of up to three simulator variables as text on the button



- **Multiplier**: multiply the variable value with this factor before displaying it
- **Decimals**: number of decimal fraction digits to display
- **Leading 0s**: pad the number to the left with '0' until this length has been reached. Example: a value of 4 makes the number 12 look like 0012.
- **Shift Y**: vertical position in pixels where the text is to be displayed
- **Split Text**: when this is set to "yes", the text will be split into multiple lines using the space characters in the string.
  Min Width: minimum width in characters that a line should have. When set to 0, the string is split at every space.
  Stretch Split: distribute the split text evenly across the height of the button
- **Background Image**: Use 144x144 pixel image files in PNG format.

Do not use the standard Elgato picture upload (the little arrow on the button icon), because this will override the text display.

Type “SteamGauge”

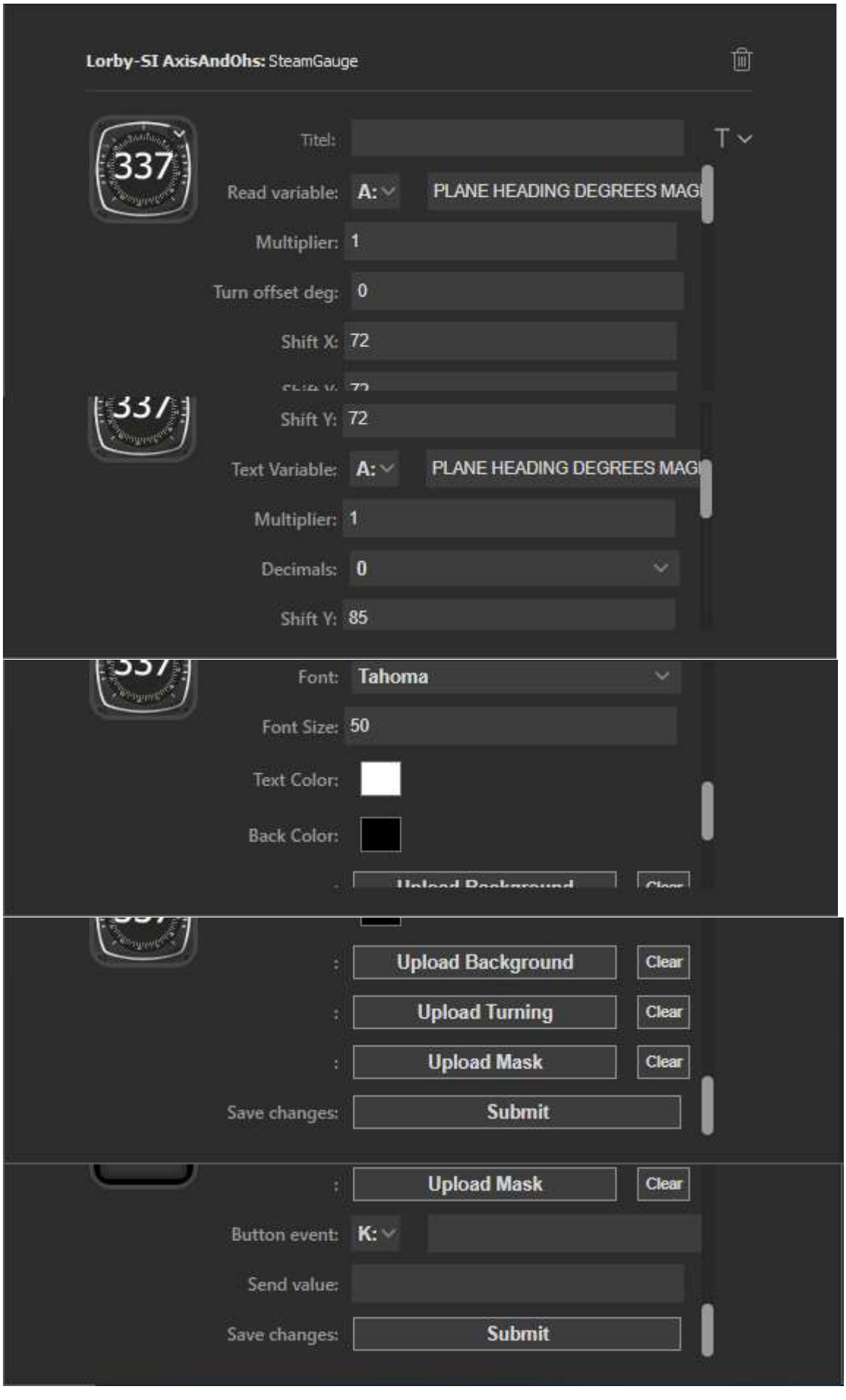
A “Steam Gauge” button displays a graphical gauge with a rotating element. An additional text can be displayed for the variable value.

The gauge consists of three layers:

- **Background (optional)**:: the static background picture
- **Turning**: the rotating part, painted on the background
- **Mask (optional)**:: another static picture that is painted on top of the other two

You can use any PNG picture with 144x144 pixel maximum size and transparency.

- **Read variable**: simulator variable to read the rotation angle from. The value must be in **Radians**.
- **Multiplier**: multiply the variable value with this factor before applying it .
- **Turn offset deg**: value (in Degrees!) to apply as the initial turn (=when the variable value is 0)
- **Shift X, Shift Y**: location of the center point for the rotation in pixels.


Lorby-SI AxisAndOhs: SteamGauge
337
Titel:
Read variable:
A:
PLANE HEADING DEGREES MAG
Multiplier:
1
Turn offset deg:
0
Shift X:
72
Shift Y:
72
Text Variable:
A:
PLANE HEADING DEGREES MAG
Multiplier:
1
Decimals:
0
Shift Y:
85
Font:
Tahoma
Font Size:
50
Text Color:
Back Color:
Upload Background
Clear
Upload Turning
Clear
Upload Mask
Clear
Save changes:
Submit
Upload Mask
Clear
Button event:
K:
Send value:
Save changes:
Submit

Type “SliderGauge”

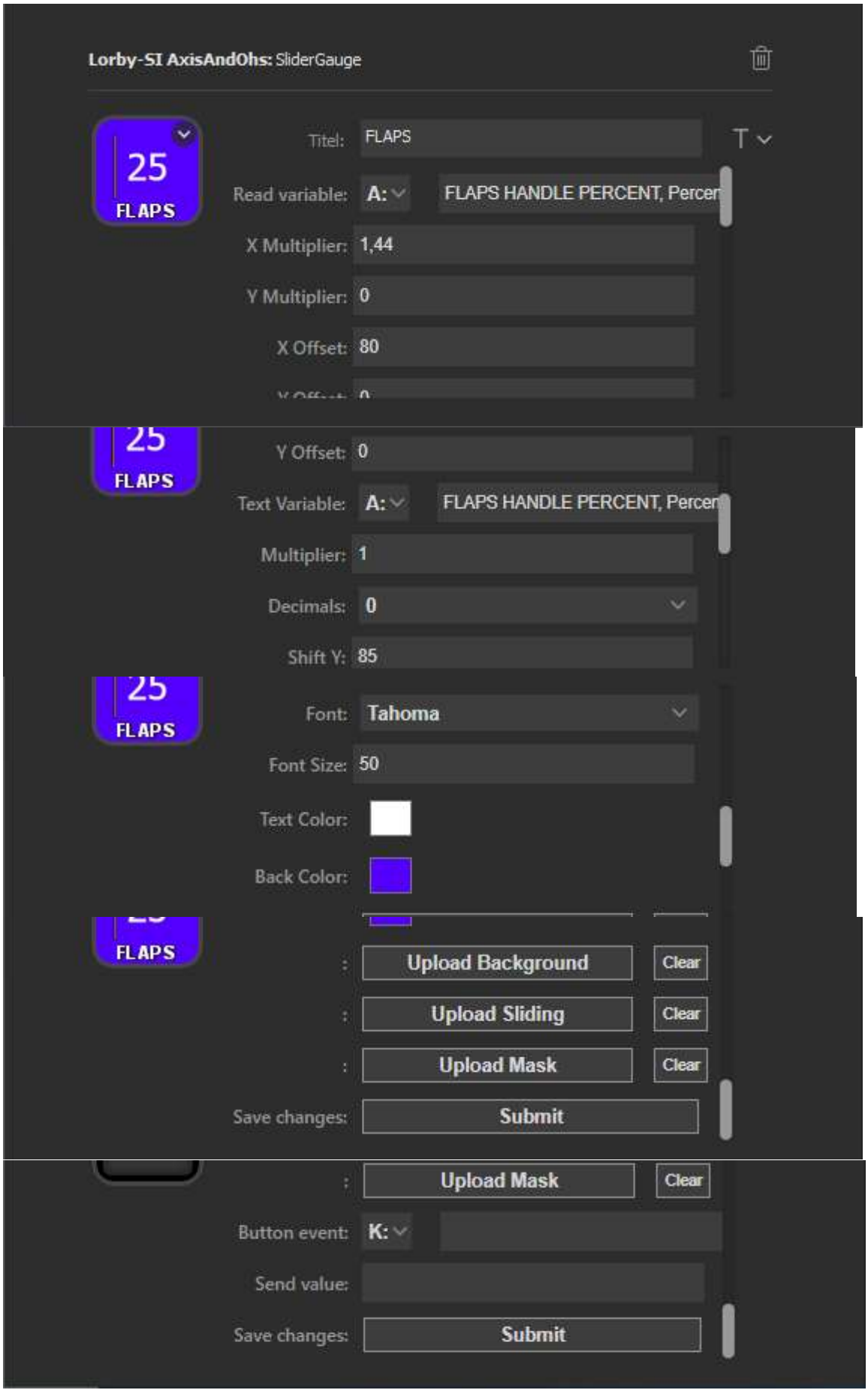
A “Slider Gauge” button uses the same layered method as the “Steam Gauge”, but instead of rotating the middle picture, it is shifted along the X and Y axis, or both.

The gauge also consists of three layers:

- **Background (optional)**:: the static background picture
- **Sliding**: the moving part, painted on the background
- **Mask (optional)**: another static picture that is painted on top of the other two

You can use any PNG picture with 144x144 pixel maximum size and transparency.

- **Read variable**: simulator variable to read the value from.
- **X Multiplier, Y Multiplier**: multiply the variable value with this factor before applying it to the associated movement axis.
- **X Offset, Y Offset**: value in pixel to apply as the initial offset (=when the variable value is 0)
- **Shift X, Shift Y**: location of the center point for the rotation in pixels.


Lorby-SI AxisAndOhs: SliderGauge
25
FLAPS
Titel: FLAPS
Read variable: A: FLAPS HANDLE PERCENT, Percen
X Multiplier: 1,44
Y Multiplier: 0
X Offset: 80
Y Offset: 0
Text Variable: A: FLAPS HANDLE PERCENT, Percen
Multiplier: 1
Decimals: 0
Shift Y: 85
Font: Tahoma
Font Size: 50
Text Color:
Back Color:
Upload Background
Clear
Upload Sliding
Clear
Upload Mask
Clear
Save changes:
Submit
Upload Mask
Clear
Button event: K:
Send value:
Save changes:
Submit

Type “MultiGauge”

The “MultiGauge” action can do all of the above at the same time.

The gauge consists of two components:

- **Multiple layers of png image definitions with transparency**: those can be static or animated. Aminations can include rotation and X/Y sliding at the same time.
- **On top the images there can be multiple text definition**: every text definition can show a different simulator variable value in a different location with its own size, color and font.

Image definitions:

To create a new image, press the “Add new image definition” button

1. Input a unique header text for the image
2. If you want the image to only be visible under a certain condition, enter a simulator variable into “Visibility variable”. Make sure that the variable returns only 0 (invisible) or 1 (visible)
   Otherwise leave this field empty
3. The “Rotation variable” works the same way as with the SteamGauge above and has the same parameters
4. The X and Y Shift variables work the same way as with the SliderGauge above and have the same parameters
5. The difference to the above is, that the MultiGauge will do all that at the same time
6. Then you decide if you want the image to fit to the button (usually when you rotate it) or not (if you create sliders, for example where an image strip represents several states of a button in each image)
7. Finally you upload the image file itself.

Once you have several layers of images, you can change their sequence with the “Layer Up/Down” buttons.

Text definitions:

To create a new text definition, press the “Add new text definition” button

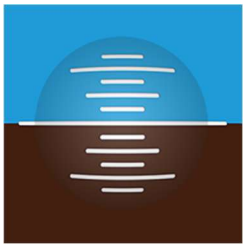
1. Input a unique header text for the text defintion

2. If you want the text to only be visible under a certain condition, enter a simulator variable into “Visibility variable”. Make sure that the variable returns only 0 (invisible) or 1 (visible)
   Otherwise leave this field empty
3. The “Text variable” defines the value that the text definition will show, and the parameters control the way that is showing it – same way as with the TextGauge action above.
4. Shift X and Shift Y set the position of the text on the gauge.

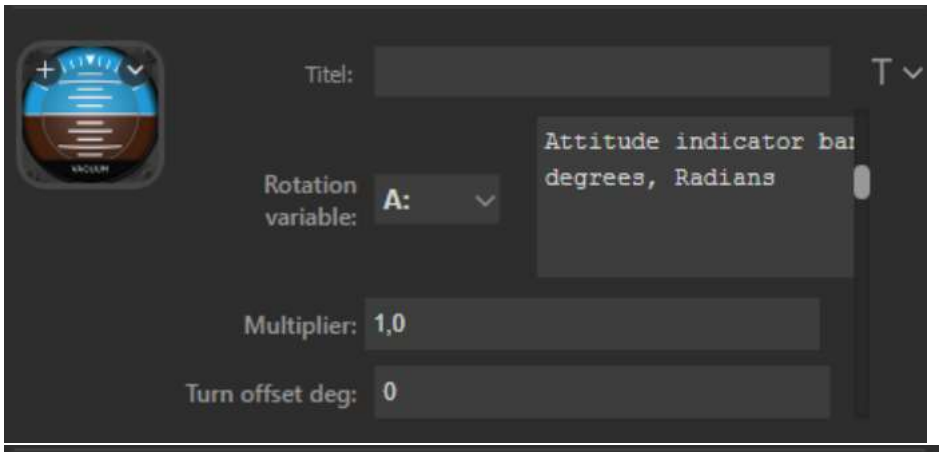
Finally, don’t forget to press the “Submit” button to save your changes.

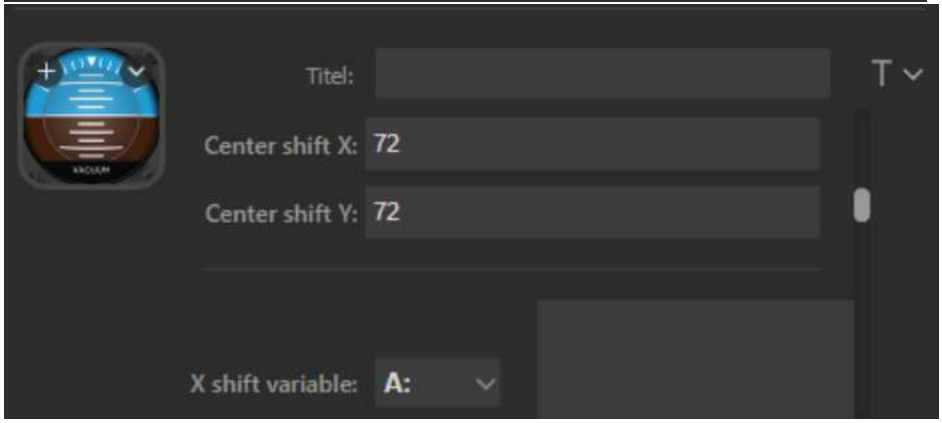
As an example, here is an artificial horizon gauge:

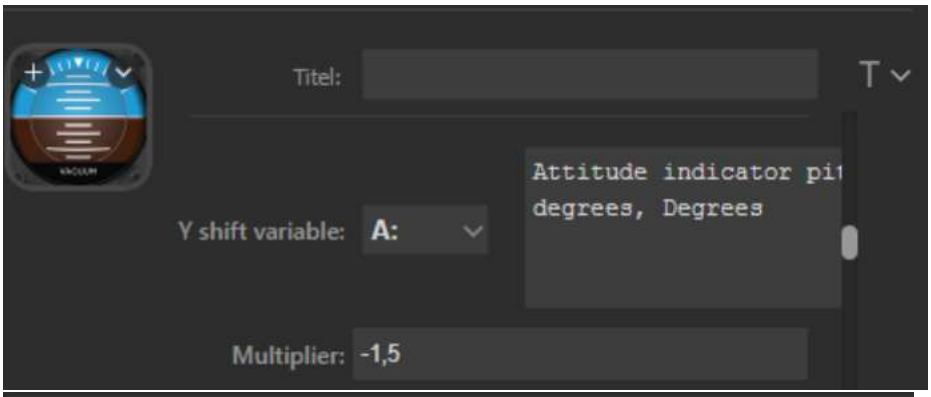
- First the background is set to show a uniform black color
- The next layer is the inner part of the AH gauge:

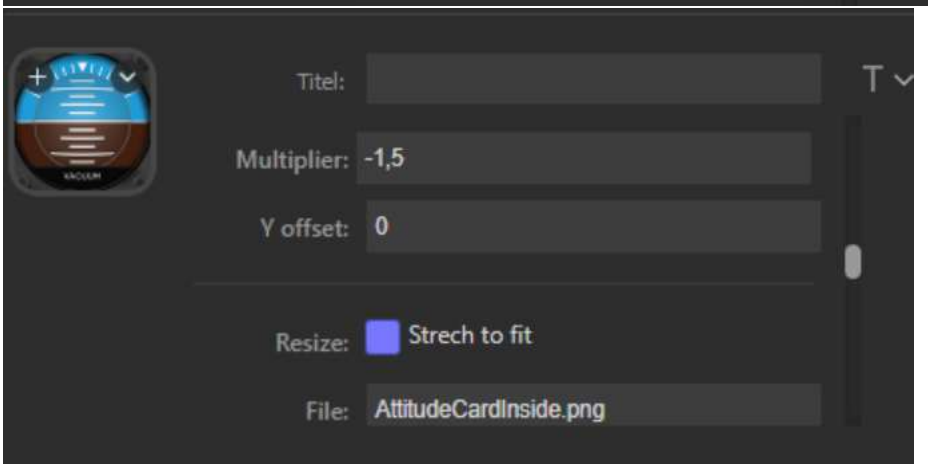


This image is rotated and shifted at the same time:

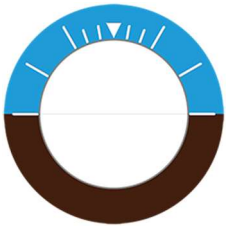


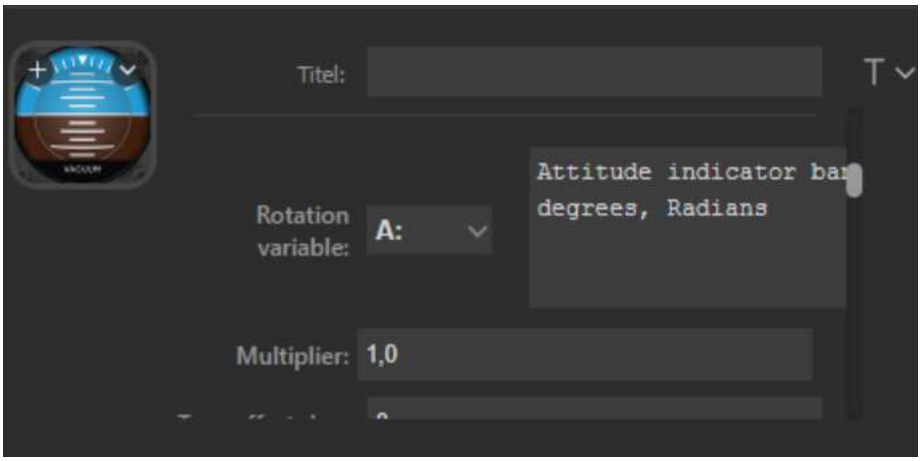


- 

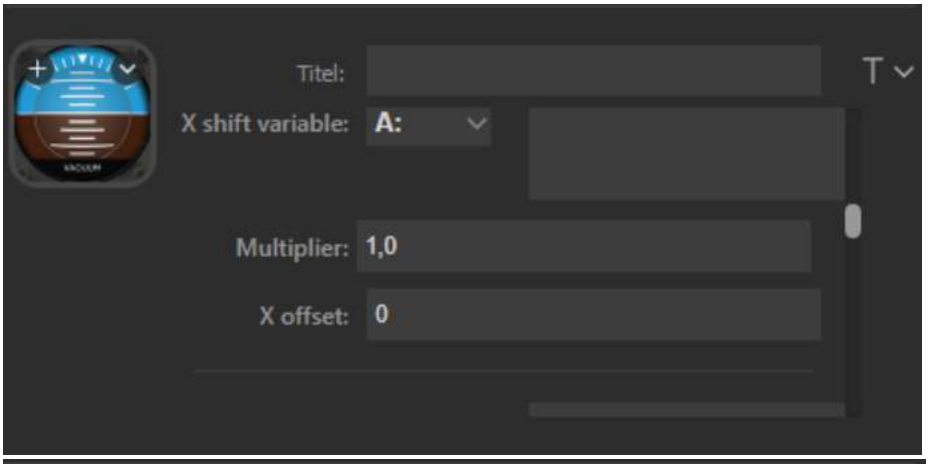
- 

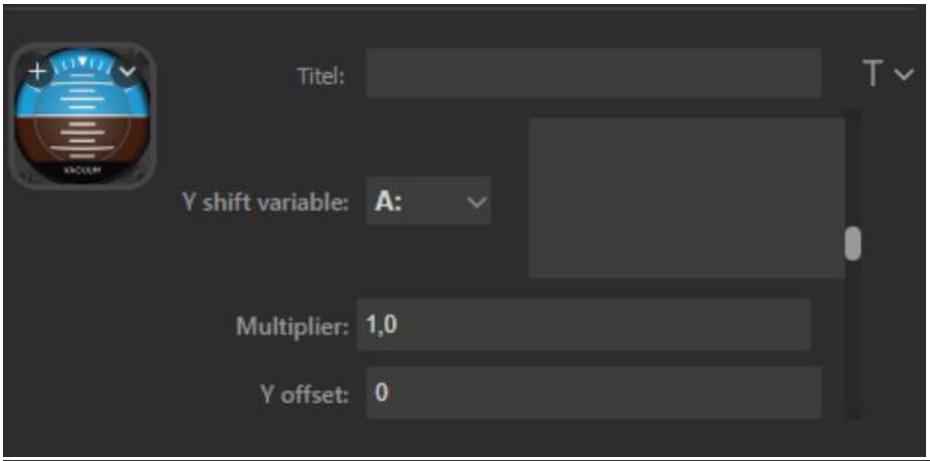
- The next layer is the outer turning circle of the AH, it only rotates

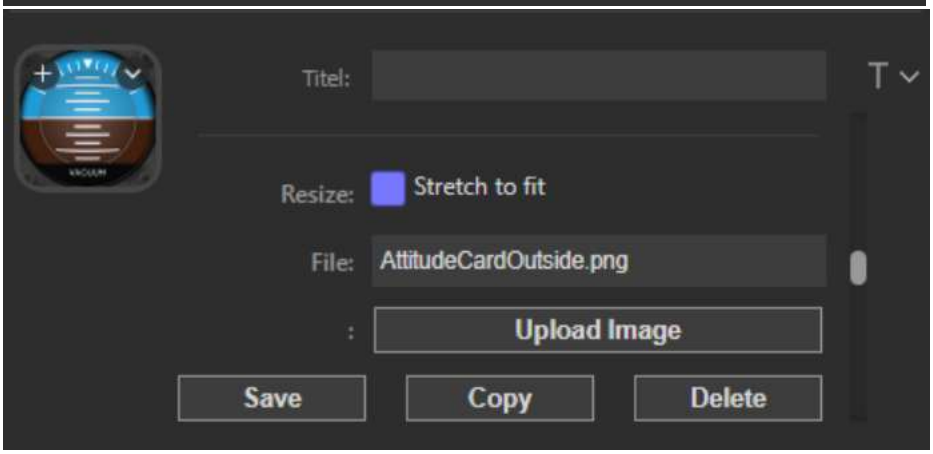




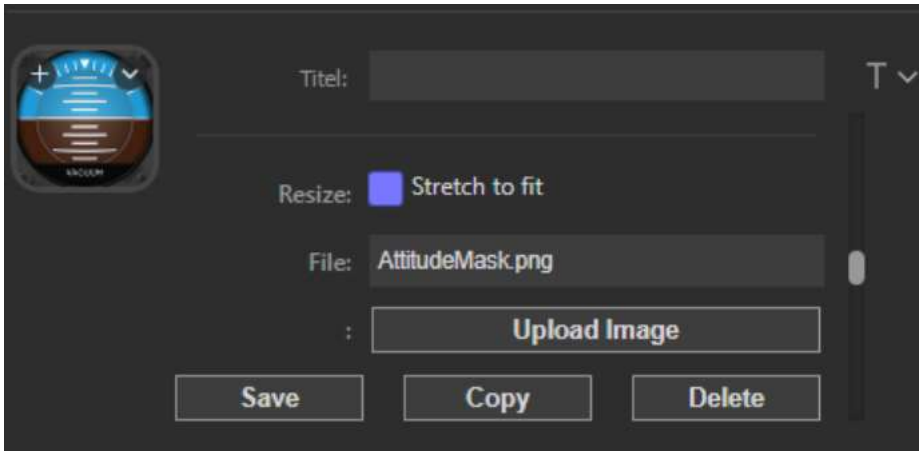
Titel:
Turn offset deg: 0
Center shift X: 72
Center shift Y: 72


Titel:
X shift variable: A:
Multiplier: 1,0
X offset: 0


Titel:
Y shift variable: A:
Multiplier: 1,0
Y offset: 0


Titel:
Resize: Stretch to fit
File: AttitudeCardOutside.png
: Upload Image
Save
Copy
Delete

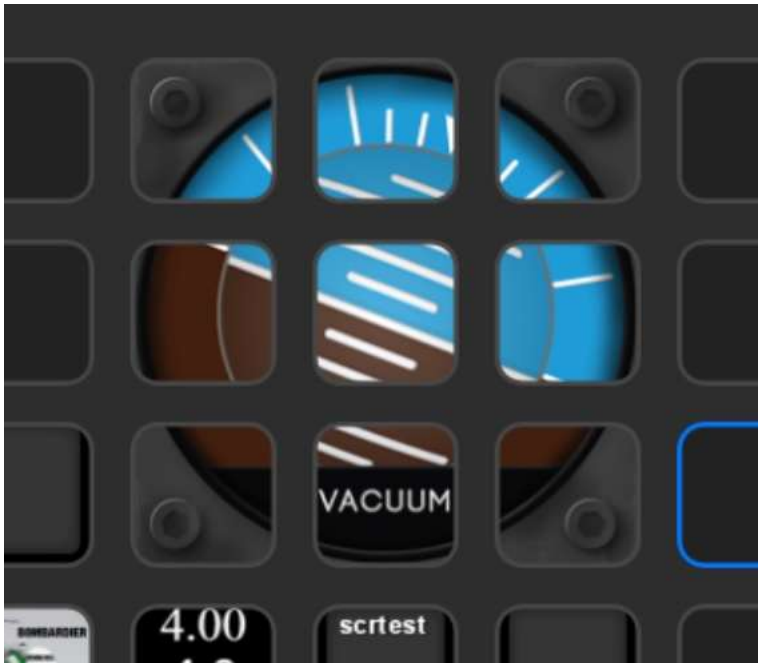
- Finally there is the covering mask of the AH. It is static and doesn't turn nor slide



To change the properties of a text or image definition, select it from the drop down box.

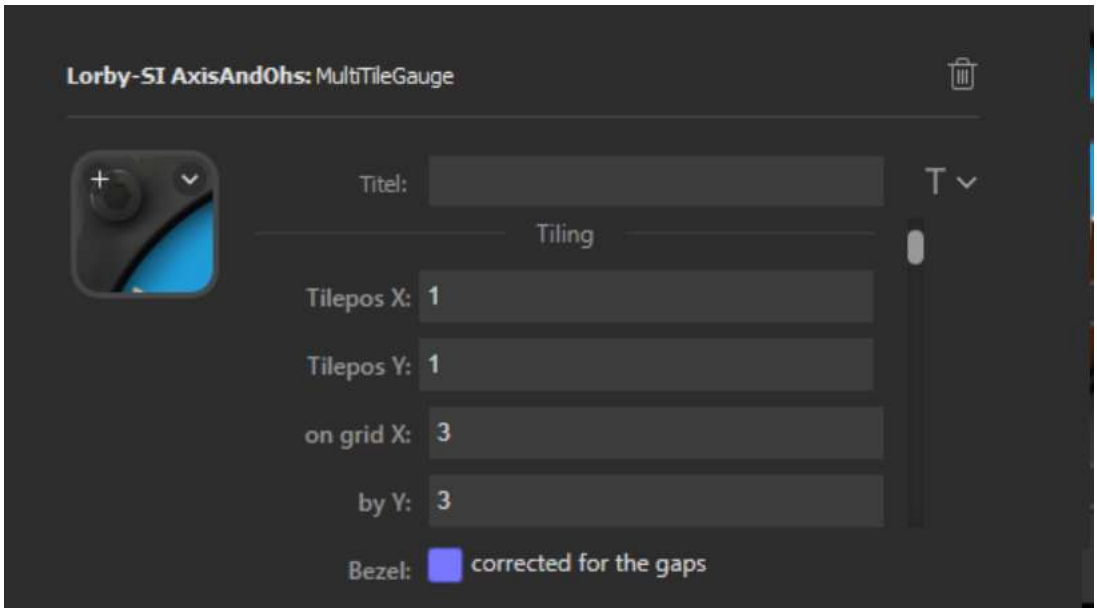
Type “MultiTileGauge”

The “MultiTileGauge” action has the same functionality as the MultiGauge, but it can be stretched over several buttons.



Each action button in the grid is a separate MultiTileGauge. Each gauge has its specific set of X/Y coordinates, starting with 1, 1 in the top left corner

The gauge has these additional parameters:



- **Tilepos X**: horizontal position of this tile in the grid (starting in the upper left corner with 1,1)
- **Tilepos Y**: vertical position of this tile in the grid (starting in the upper left corner with 1,1)
- **On grid X**: total number of horizontal tiles in the grid
- **by Y**: total number of vertical tiles in the grid
- **Bezel**: Activate this to instruct the plugin to subtract the gaps between the buttons from the image.

The best strategy to create a tiled gauge is by starting in the top left corner and setting that MultiTileGauge up correctly for the images, turning and moving elements. Once that is complete, copy this action on each tile and simply adjust the Tilepos X/Y parameters (don't forget to press "Submit").

Be mindful that you are dealing with an image that will be stretched over multiple buttons, so adjust center of rotation and shift parameters accordingly.

One button has a size of 144x144 pixel, the bezel/gap is set to 48 pixel.

Example: with the attitude gauge as MultiGauge, the center of rotation was 72x72 (the center of the 144x144 button) and the shift multiplier for the artificial horizon was -1.5. When this gauge is stretched over a 3x3 grid with bezel correction, then the center of rotation is (144*3 + 48*2) / 2 = 264
(the center of three buttons with 144px plus two gaps with 48px)
Likewise the multiplier is -5.5, as the total height of the image is 528px instead of 144px.

If a parameter is wrong, instead of correcting all the buttons, delete them all except for the one at 1,1 – then correct the parameter in this action only, copy it again on the other tiles and adjust the Tilepos X/Y.

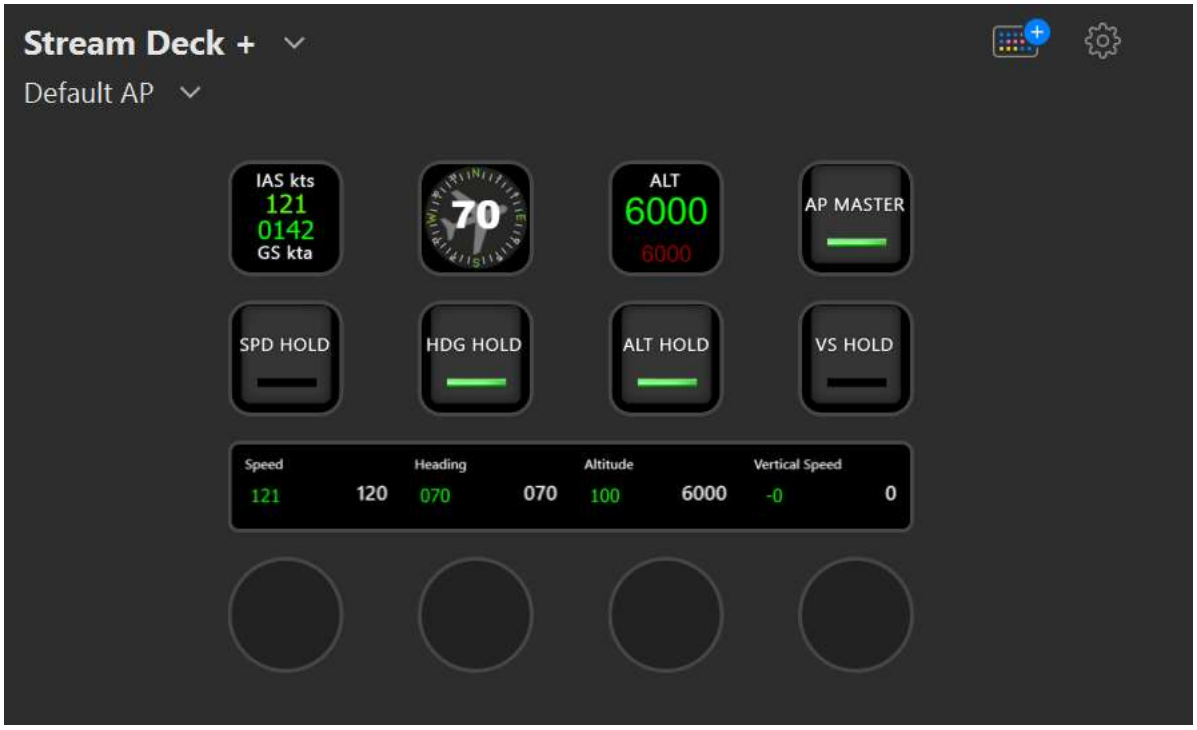
Types “Rotary Encoder” and “Rotary Encoder Canvas”
(Stream Deck Plus only)

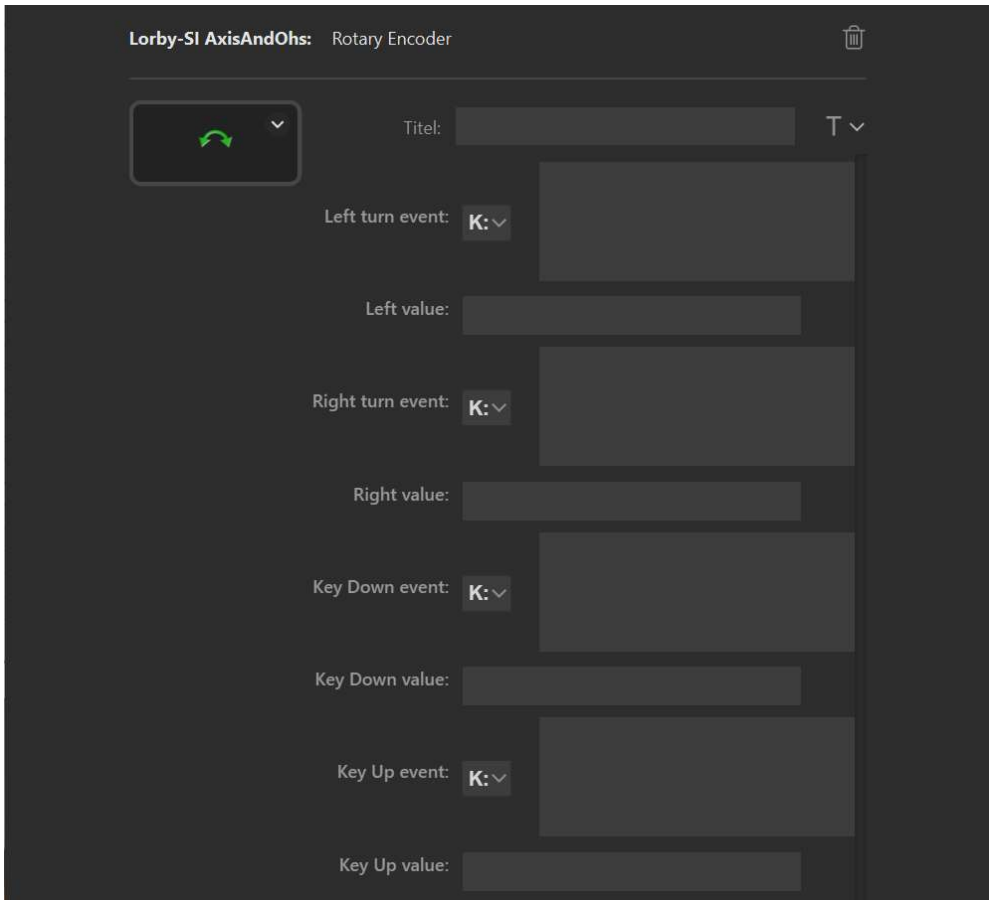
The “Rotary Encoder” is only available on the StreamDeck “Plus”.

It has four actions: left turn, right turn, click and a tap on the display above it.

- The display of the “**Rotary Encoder**” can show a title, two numerical values and a slider. All elements are optional.

- The display of the “**Rotary Encoder Canvas**” works the same way as the “MultiGauge” described above. You can layer any number of image and text definitions over each other, covering the full 200x100 Pixel display in the touch strip above the Encoder





- **Left turn**: event and value for the left turn
- **Right turn**: event and value for the right turn

- **Fast turn speed**: turning speed that is seen as a “fast turn”
- **Left fast turn**: event and value for the left fast turn
- **Right fast turn**: event and value for the right fast turn

- **Key Down event**: event and value when the encoder is pressed
- **Key Up event**: event and value when the encoder is released

- **Touch Tap even**: event and value for tapping on the value display

- **Long press event**: long press is picked up only for the encoder, not the touch screen

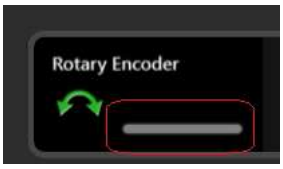
Additional properties of the “Rotary Encoder”

- **Value variable**: is shown to the right on the display

- **Icon:** is shown below the title

You can upload image definitions for the icon display, controlled by a visibility variable. This is the same principle as with the MultiGauge above, but the icon pictures have to be 144x144 and can’t be moved or repositioned.
On top of the icon you can draw the value of a variable. The text color can be changed dynamically with the “Color Variable” option. The color must either be a string LVar or a script returning a color string. The format of the string is ‘#rrggbb’

- **Bar variable**: controls the value of the “progress bar”. The value must be between 0 and 100.



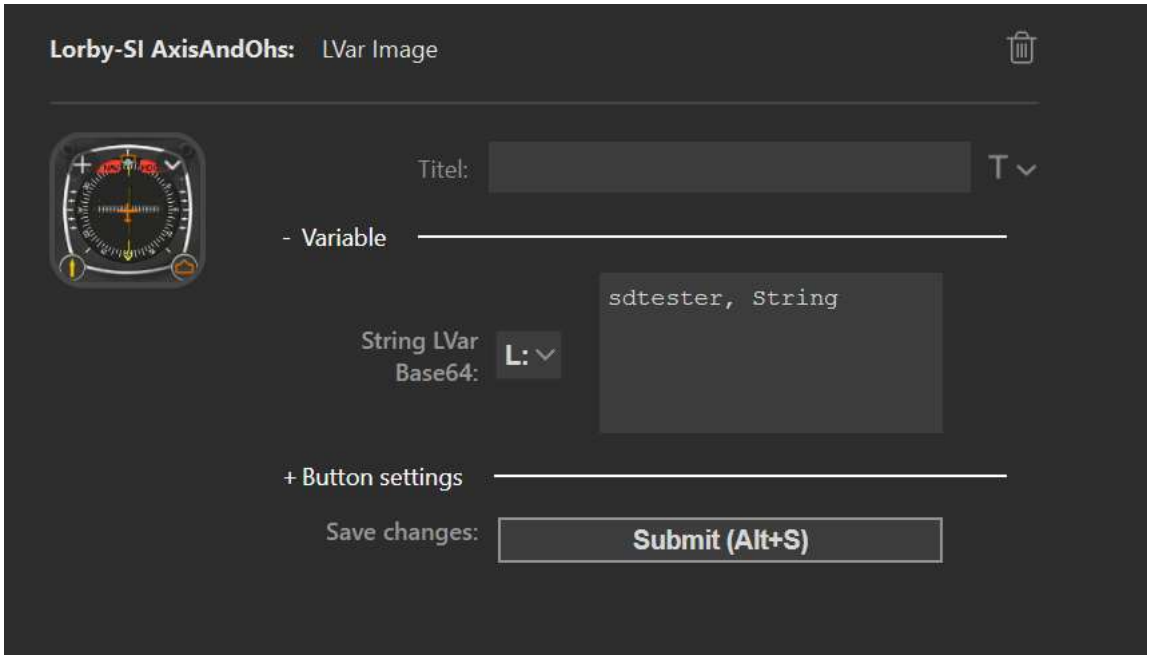
All variables are optional. When you don’t supply a variable name, the corresponding element will not be displayed.

Additional properties of the “Rotary Encoder Canvas”

See the description of the “Multi Gauge” – handling and properties are identical

Type “LVar Image”

This is an AAO specific action that allows you to change the image on the action button with an LVar from within AAO, using scripts that provide image data in Base64 encoding.



The only parameter that this action needs is a String LVar containing the Base64 encoded image.

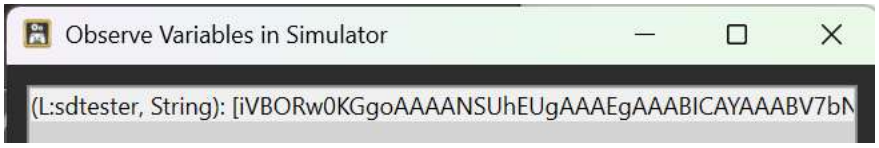
Typically, you would load this image from within an AAO script like this:

```
(LOADIMAGE:hidtest.png|sdtester)
```

This command loads the file \Documents\LorbyAxisAndOhs Files\hidtest.png into the LVar (L:sdtester, String). This variable is then used by the action to paint the image on the button.

Note 1: this change is permanent. When you set the image, it will stay this way until you apply a new one.

Note 2: in case you want to create your own Base64 strings – the action expects the pure data, in png format, without the header:



Please read up on the LOADIMAGE command and its variants in the AxisAndOhs manual.