Streamdeck Watson Control

Contact Info

lpower@sbgtv.com – Liam Power - 207-228-7650

Purpose: This script allows you to control your Watson closed captioning systems via an [ElGato Streamdeck](). Functionality includes starting captioning, stopping captioning, and viewing the current status of the captioner.

Operation:

1. Panel usage – See image on next page
   a. Closed Caption Status Indicator: Illuminates green when captions are running, as seen on Watson 1 on the right, and red when captions are not running, as seen on Watson 2 on the left.
   b. Start Closed Captioning: Starts the closed captioning on the Watson, with the numbers corresponding to the Watson unit. If captioning is already running, this has no effect.
   c. Stop Closed Captioning: Stopss the closed captioning on the Watson, with the numbers corresponding to the Watson unit. If captioning is not running, this has no effect
   d. Exit: This exits out of the python script, leaving the 'Launch Watson Control' icon in the middle active. Useful for debugging or restarting in case of issues.
   e. Launch Watson Control: This launches the Python script that controls the system. If the program exits cleanly, this button will remain after exit, and if you've also installed the ElGato Streamdeck management application, you can set it up to ensure it is set to that location through there as well.
   f. Station Logo: This has no functionality beyond looking pretty.
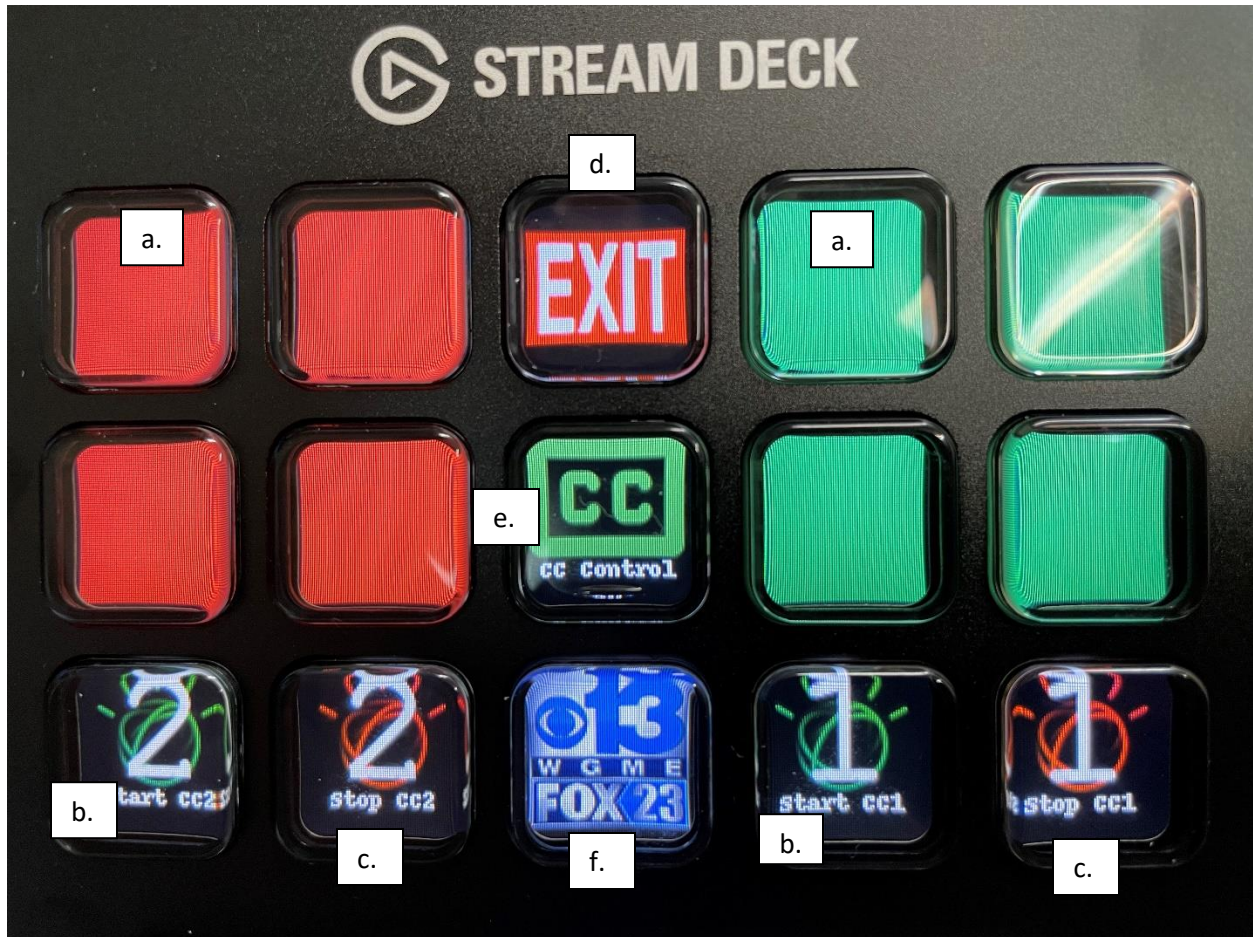
Installation
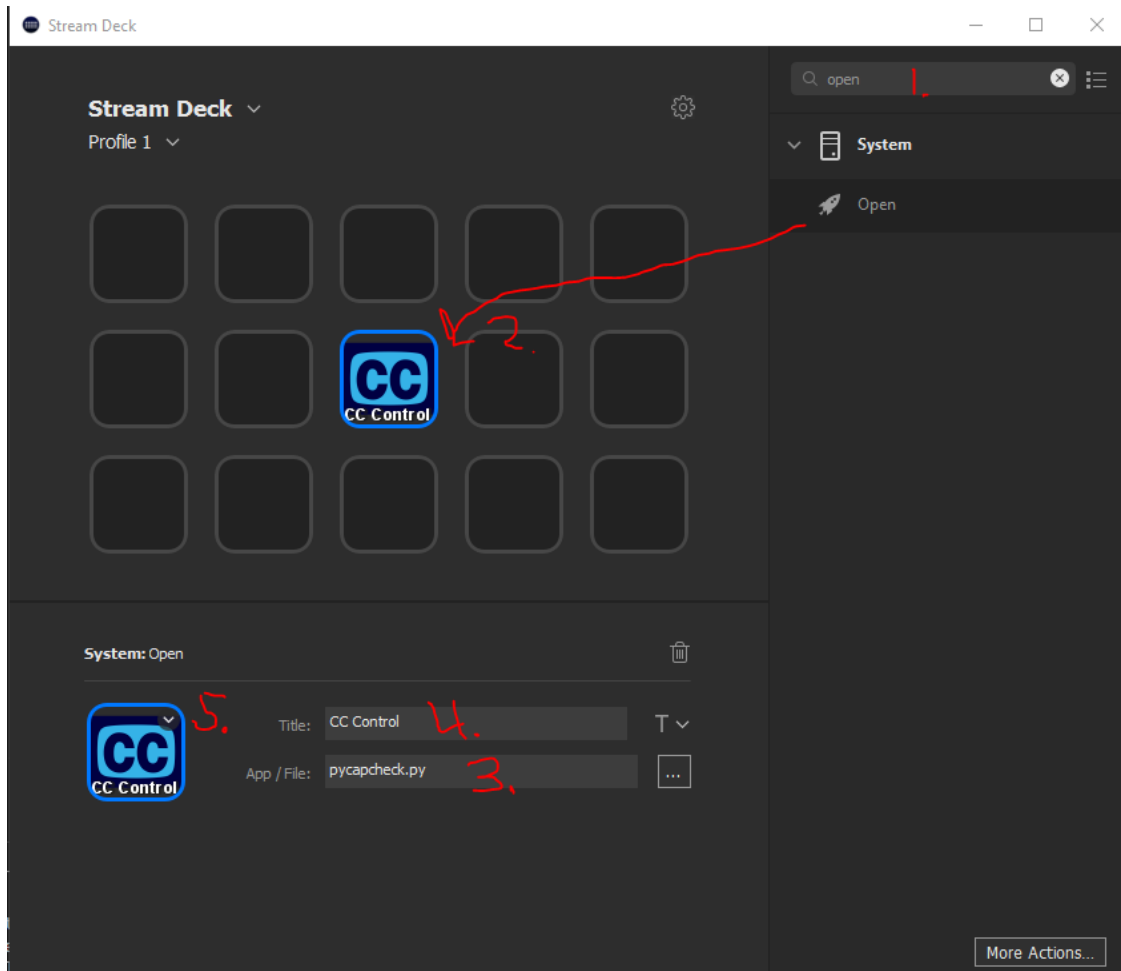
This has several dependencies. Pillow and HIDAPI have installation instructions baked into the Streamdeck library instructions.:

- Python 3.9 – May work with older Python 3 versions, just untested.
- Python-Elgato-Streamdeck Library – This can be installed via npm. Documentation can be found at the link. This allows easy interaction with the Streamdeck.
- Pillow – Also installable via npm. This is used to make the images for the key tiles.
- HIDAPI – Enables interaction with HID devices. Used by the Streamdeck library to communicate with the deck. If using Windows, ensure you use the correct 64/32 bit.
- Requests – This handles pulling from the Watson API.
- (Optional) ElGato Streamdeck Software – This is not strictly necessary, it is simply useful for ensuring the launch button is always on the deck.

Ensure all the above are installed. You should also have a folder with the script and an 'Assets' subfolder that contains 9 PNG images. The script is relative, so place the folder wherever you like before doing the following:

1. Variable Setup – This will configure the Streamdeck for your environment, and can be done in any text editor of your choice. You should see the section starting with 'Image Locations' near the top of the file. For the key sections, note that the keys are in order left to right and top to bottom, starting with 0, so the top left key is 0 in the above image, and the bottom right one is 14. Most key types only allow one of those keys at a time, with the exception of the status keys.

    a. ASSETS_PATH – This is the folder in the same directory as the script that stores the images. To change the name of the folder used, replace the Assets inside the quotes with the new folder name.

    b. exit_key_index – This is the location of the 'Exit' key.

    c. cc1on_key_index – This is the location of the key to turn on Watson 1.

    d. cc1off_key_index – This is the location of the key to turn off Watson 1.

    e. cc2on_key_index – This is the location of the key to turn on Watson 2.

    f. cc2off_key_index – This is the location of the key to turn off Watson 2.

    g. cc1_key_index – This is the location of the keys that show the status of Watson 1. Note that you can add or remove keys showing the status by changing the number of comma-separated numbers inside the brackets.

    h. cc2_key_index – This is the location of the keys that show the status of Watson 2. It can be edited like the Watson 1 status keys.

    i. launch_key – This is the location of the key that launches the script.

    j. stat_key_index – Any key added to this list will be created without any text labels, just the image.

    k. brightness – The brightness of the deck, from 0 to 100. This and the Streamdeck application will override each other, so if using both, ensure they're set to the same brightness.

    l. deckid – This is the most important one to change! When you run the script, it will show you a list of all the connected Streamdecks, but will only connect to the deck with the id set here. This means you'll have to run it once, copy the deck id into the script, and then run it again to actually work. The image below shows you where the id to copy is. Replace *only* the part between the quotes in the script, ensure the r remains outside them.

    m. cc1_host – This is the Name/IP of the first Watson.

    n. cc2_host – This is the Name/IP of the second Watson.

2. (Optional) Streamdeck Application - You should now be able to run the script manually via the command line, and from the panel thereafter. However, if you want to have the panel initially populate with the launch button, you can do so using the Streamdeck application.

    a. First, search for 'open' in the search box.

    b. Drag 'open' onto the button you plan to use as the launch button. By default, this is the middle one.

c. Click the three dots to the right of the box, and locate the script py file.
d. Fill in the label with what you want it to be called.
e. Click the arrow over the image, choose from file, and select the SWCLogo image.

Known Issues:

- Currently only allows for two Watsons.
  - Will be fixed in the dynamic allocation update.
- Currently only allows for the 15 key Streamdeck.
  - Will be fixed in the dynamic allocation update.

Future Plans:

- Dynamic sizing/layout/Watson count
- More user-friendly variable entry

License:

Python-Elgato-Streamdeck used under MIT License:

© Copyright 2020, Dean Camera

Permission to use, copy, modify, and distribute this software
and its documentation for any purpose is hereby granted without
fee, provided that the above copyright notice appear in all
copies and that both that the copyright notice and this
permission notice and warranty disclaimer appear in supporting
documentation, and that the name of the author not be used in
advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

The author disclaims all warranties with regard to this
software, including all implied warranties of merchantability
and fitness. In no event shall the author be liable for any
special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether
in an action of contract, negligence or other tortious action,
arising out of or in connection with the use or performance of
this software.

The Python Imaging Library (PIL) is

  Copyright © 1997-2011 by Secret Labs AB
  Copyright © 1995-2011 by Fredrik Lundh

Pillow is the friendly PIL fork. It is