**Equipment List**

* Laptop/Desktop Computer
* Pure Data extended/vanilla
* Standard PlayStation 4 controller
* 1x Mini USB to USB lead to connect the controller to the Computer

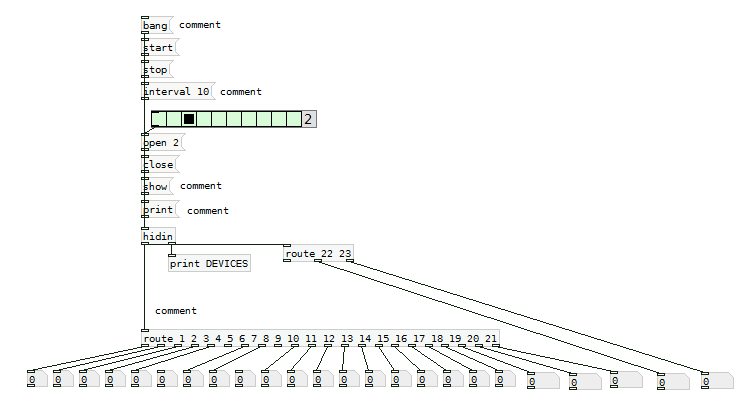
**Intro**

The GitHub page will give you all the tools to create a PS4 controller arpeggiator. This was created to help me perform arpeggios live with my band Cause to Effect and be able to change the tempo in accordance with the drummer’s tempo. The patch includes a tap tempo, a simple reverb, a simple delay, multiple scales to choose from and a counter which can be adjusted using the controller. The counter can be adjusted by selecting how many notes in each arpeggio and the direction of the arpeggio which will be either ascending, descending or a mixture of the two.

**Development Diary/Operation manual**

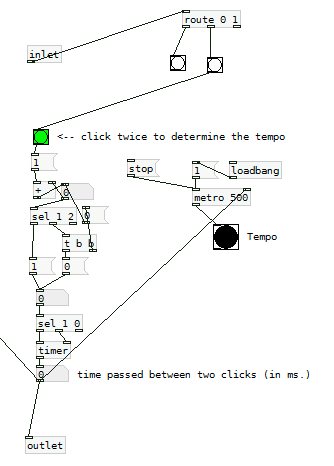
The first step for creating this patch was to research other similar patches to attain whether what I was planning had either already been done or had been done better.

I found a large amount of controller based instruments including drum patches and guitar patches which were using the same hardware and some of the same software. Which for this patch the main software will be Pure Data. The second step was to start building the patch. I started by using a HID object (human interface device) to detect the PS4 controller and extract the values and button presses from the controller and transpose them into a readable language within pure data.



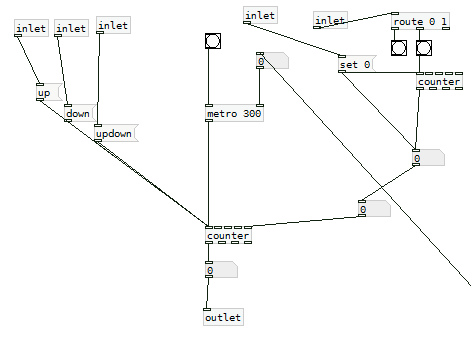
*Fig - 1*

As can be seen from the picture above (fig -1) this is a basic Hid object which for windows operators like myself has to be a Hidin object which works the same of the Mac Hid equivalent. The grid at the top of the patch after the band and start objects relates to the amount of inputs PD is reading from. One clicking the print devices button a list of available devices will appear in the PD window with a number attached to each. Once the correct number for your controller has been selected and the open message has been selected PD will then start reading the incoming information from the controller device. As can be seen from Fig – 1 there are 23inputs coming from the controller which can be patched to any function within PD.

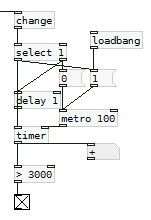
The next step was to start building all of the controls for the arpeggiator including the counter, tap tempo, FX and a few other patches that didn’t work out in the end. I start with the tap tempo which can be found within the counter sub patch within the included files.

*Fig - 2*

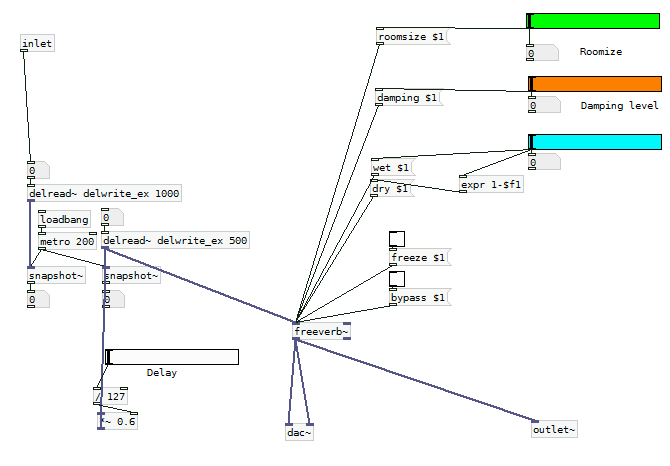
The tap tempo is very easy to use and can be triggered by depressing the green bang at the top of the patch. I routed the L3 (click left analogy stick in) for this task.



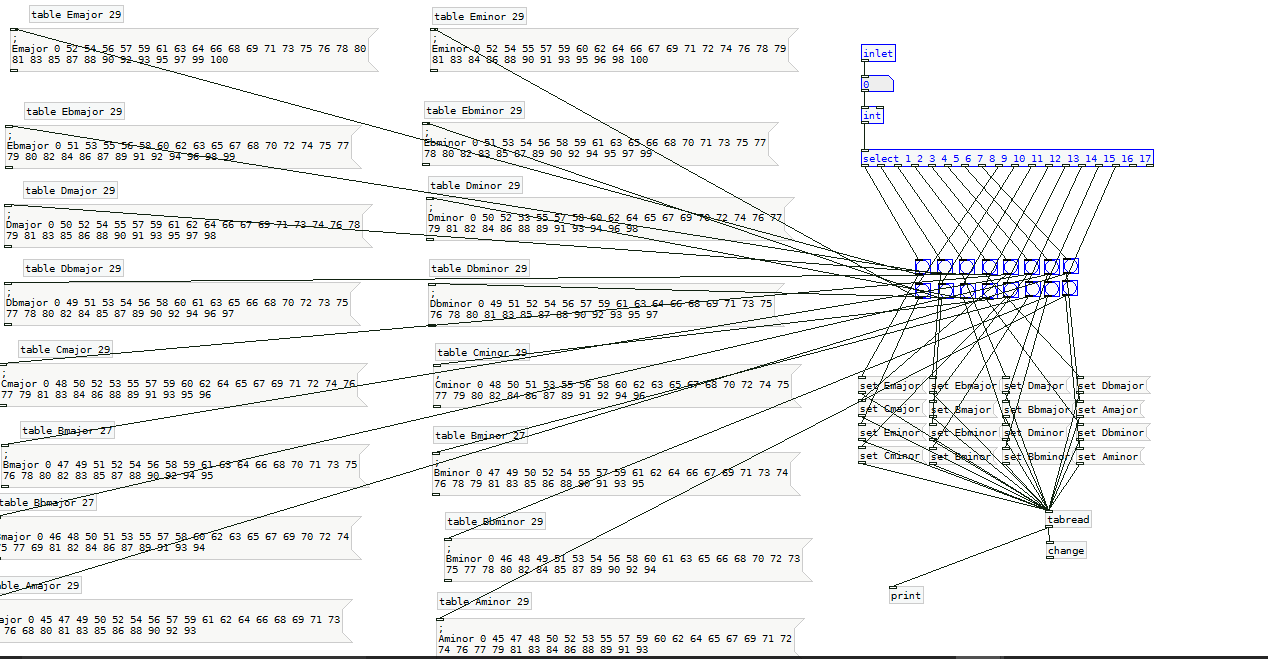
*Fig - 3*

Fig-3 above shows the counter section of the arpeggiator. Operated by using the X button on the controller. Click the button down as many times as needed until you reach your desired. The section to the upper left of Fig-3 controls the direction of the arpeggiator. I used the directional section of the controller to control this UP for ascending DOWN for descending and right for up and down.

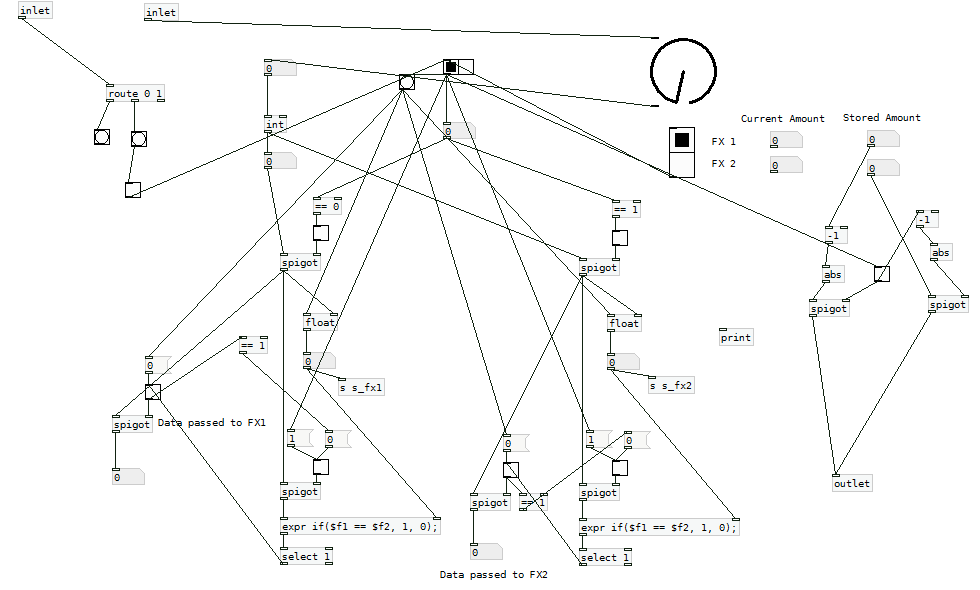
*Fig - 4*

Fig-4 is a part of the patch that relates to the counter. If the X button is held down for more than three second a band is send to a message box to reset the counter back to zero which means one button is used to control this freeing up other controls for other options within the patch.

*Fig - 5*

The next modular patch to be built was the effects section. Above in Fig-5 is a simple delay and reverb which are situated just before the output. This are controlled using the square button for the reverb and circle button for the delay. The more presses the patch receives the high the wet dry signal goes on the effects. They can also be reset to zero using the triangle button.

*Fig - 6*

The final piece of the puzzle is the scale select function. This is activated using the right analogy stick. The analogy stick is patched into a slider object with a scale of one to seventeen (one for every scale present). As the analogy stick is turned from directly up and around the sweep the different scales can be selected.

*Fig - 7*

Fig-7 shows a part of the patch which is included in the documents but sadly is not used at this moment in the finished patch but I wanted to keep it in as it can be used to improve and develop the patch at a later date. This section uses a sequence of spigot objects and IF objects to select one note and have that played while a second note is being selected or searched for without it being played out of the main output. This was originally used to select the note for the arpeggio to use as a start note and worked off the left analogy stick using the dial seen at the top right of FIG-7 as a visual representation of the analogy stick.

The final task was to route all of the patches together and get them to work in conjunction with each other. I did this by have the scales sub-patch go into a make note object then into a phasor object which is the main sound source for the patch then into the effect sub-patch which in turn is routed into the output. The counter and tap tempo are routed into the scale section patch which helps in the selection of the notes and scales used.

**Room for Improvements**

Given more time I would like to incorporate more sounds and effects into the patch to make it a more versatile instrument. A floating note using the left analogy would also be a powerful addition, this could be used through a separate output and be play alongside the arpeggio to give the user a more versatile and unique sound.