**Assignment 4: Vibrato Plug-in**

Amruta Vidwans and Milap Rane

We answer every question mentioned in the assignment one by one, with the line of code and the file where each of these were implemented:

1. Modify PluginProcessor.cpp/h

1. to instantiate your vibrato class with a default of 2 channels

Vibrato.cpp line 22.

1. to set samplerate in prepareToPlay

PluginProcessor.cpp line 57.

1. to call your process function from processBlock

PluginProcessor.cpp line 89.

1. to set and get parameters from set/getParameters (you don't want that possibly called during process, but you don't want to use critical sections, etc, either - what is the simplest solution?)

PluginProcessor.cpp line 149 and line 155 .Here, we have been able to use setParameters while the process function is running, by adding a flag in the process function for change in the slider values. If there is change, it implements a set function in the Vibrato class for the parameter changed for the slider. This is the simplest solution.

1. to implement processBlockBypassed (remember that the user will want to hear something immediately after disabling bypass)

PluginProcessor.cpp line 93. We are also implementing a toggle button button to bypass the Process in our code.

1. remove all synthesizer/midi keyboard related members from the code

Synthesizer/MIDI related members have been removed.

1. correctly change all remaining functions from the demo plugin to reflect the properties/requirements of your plugin

This has also been done.

2. Modify PluginEditor

1. implement the GUI controls above and map them to the parameters setting/getting functions [15]

Whenever the slider value was changed , we used the setParameter function of the plug-in, to reflect the change in the Vibrato class. Also, we have used the getParameter function, in the constructor to retrieve the last value of the slider.

1. remove all synthesizer/midi keyboard related members from the code [5]

Synthesizer/MIDI related members have been removed.

1. keep the GUI code as 'stupid' as possible: no calculations (except things that are only needed for the GUI, e.g. string conversions): that is all done in your AudioProcessor! The GUI is only for displaying things!
2. add a label to the GUI as the 'plugin title' of your choice

This has been done.

We have implemented a GUI with two horizontal sliders, one to change the modulation frequency, and modulation width. The ranges are set to be 5-15Hz for modulation frequency, and 5-10 ms for modulation width, as given in DAFx book’s vibrato description. The GUI displays the changed values and it’s units through a text box, as well as a pop-up display. The units displayed are Hz and sec, for modulation frequency and modulation width respectively.