Audio Connections & Memory

AudioConnection objects route the output of an output of any audio object to the input of another. Useful audio applications usually require several audio objects and of course connections between them.

Audio data is stored in 128 sample blocks, which you allocate with AudioMemory. Functions described below allow you to monitor the audio library's usage of this memory.

Usage

AudioConnection myConnection(source, sourcePort, destination, destinationPort);

Route audio data from the source object's sourcePort, to the destination object's destinationPort.

AudioConnection myConnection(source, destination);

Route audio data from the source object's output 0, to the destination object's input 0. This shorter form is convenient when using objects with only a single input and output.

AudioMemory(numberBlocks);

Allocate the memory for all audio connections. The numberBlocks input specifies how much memory to reserve for audio data. Each block holds 128 audio samples, or approx 2.9 ms of sound. Usually an initial guess is made for numberBlocks and the actual usage is checked with AudioMemoryUsageMax().

AudioMemoryUsage();

Returns the number of blocks currently in use.

AudioMemoryUsageMax();

Return the maximum number of blocks that have ever been used. This is by far the most useful function for monitoring memory usage.

AudioMemoryUsageMaxReset();

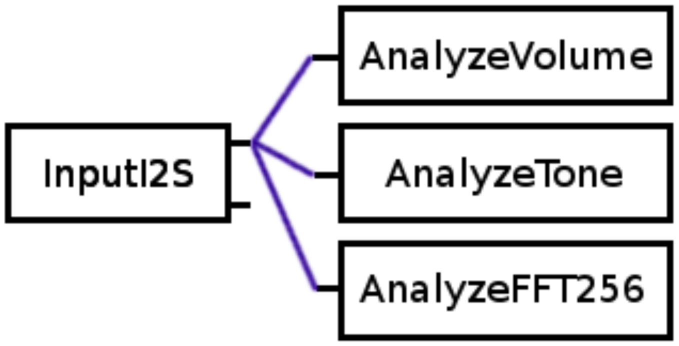
Reset the maximum reported by AudioMemoryUsageMax. If you wish to discover the worst case usage over a period of time, this may be used at the beginning and then the maximum can be read.

AudioConnection objects do not have any functions. They are simply created in your sketch, after the audio objects, to define the data flow between those objects.

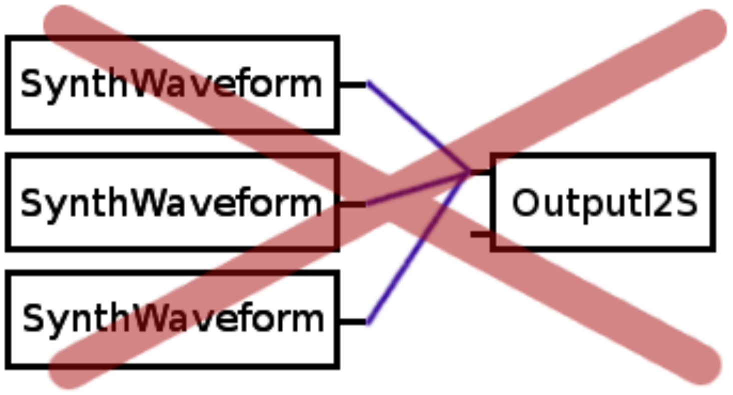
Connection Rules & Guidelines

Normally a single output connects to a single input.

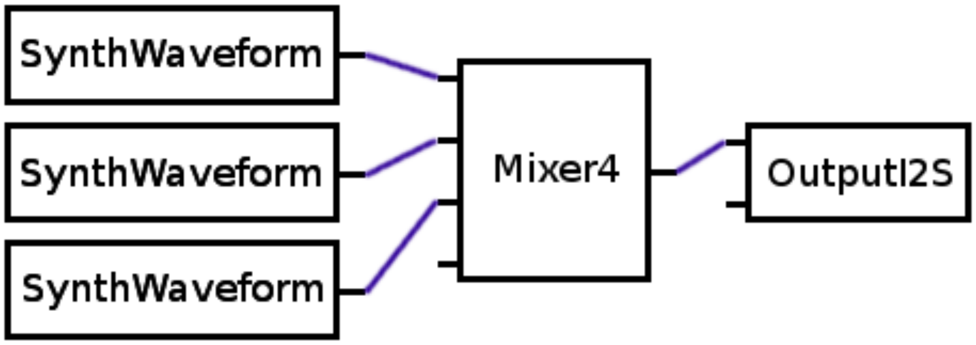
You may send the output from one object to as many inputs as you like. The audio library efficiently handles objects sharing the output of an object.



However, each input can receive only a single connection. If you attempt to connect many outputs to a single input, only the last connection will work.



To feed multiple outputs to a single input, use the Mixer object to combine them to a single output.



Audio objects should be created in the order data is processed, inputs, playback and synthesis, then effects, filters, mixers, and lastly outputs.

Connections are most efficient when made from an earlier object (in the order they are created) to a later one. Connections from a later object back to an earlier object can be made, but they add a 1-block delay and consume more memory to implement taht delay.

Example Program

TODO: an example showing varying memory usage...