ofxTonic

Frequency Modulation

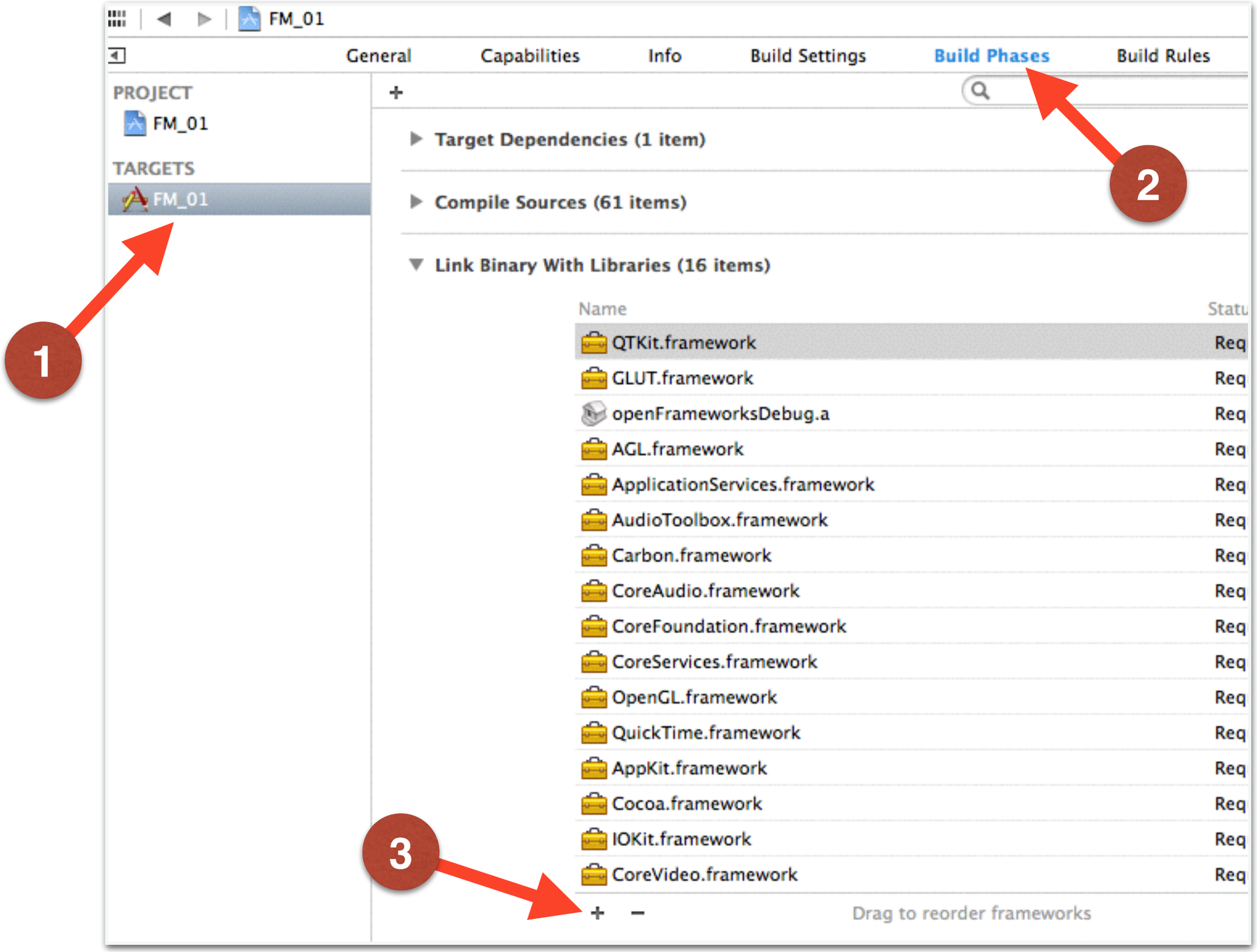
setup

1. ofxAddons : ofxTonic
   https://github.com/TonicAudio/ofxTonic

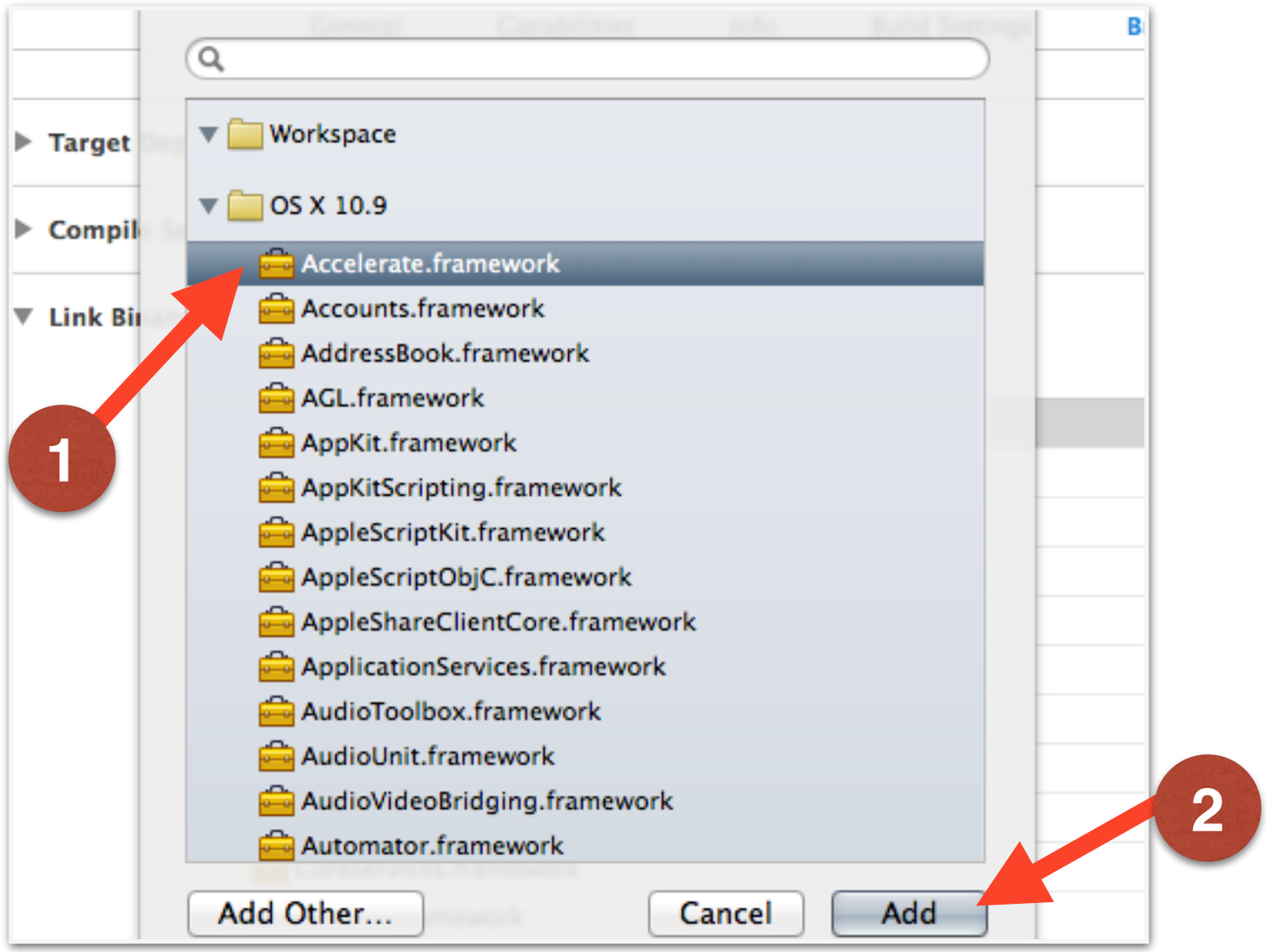
2. Accelerate.frameworks

*. TonicAudio (Andere Beispiele)
   https://github.com/TonicAudio/Tonic

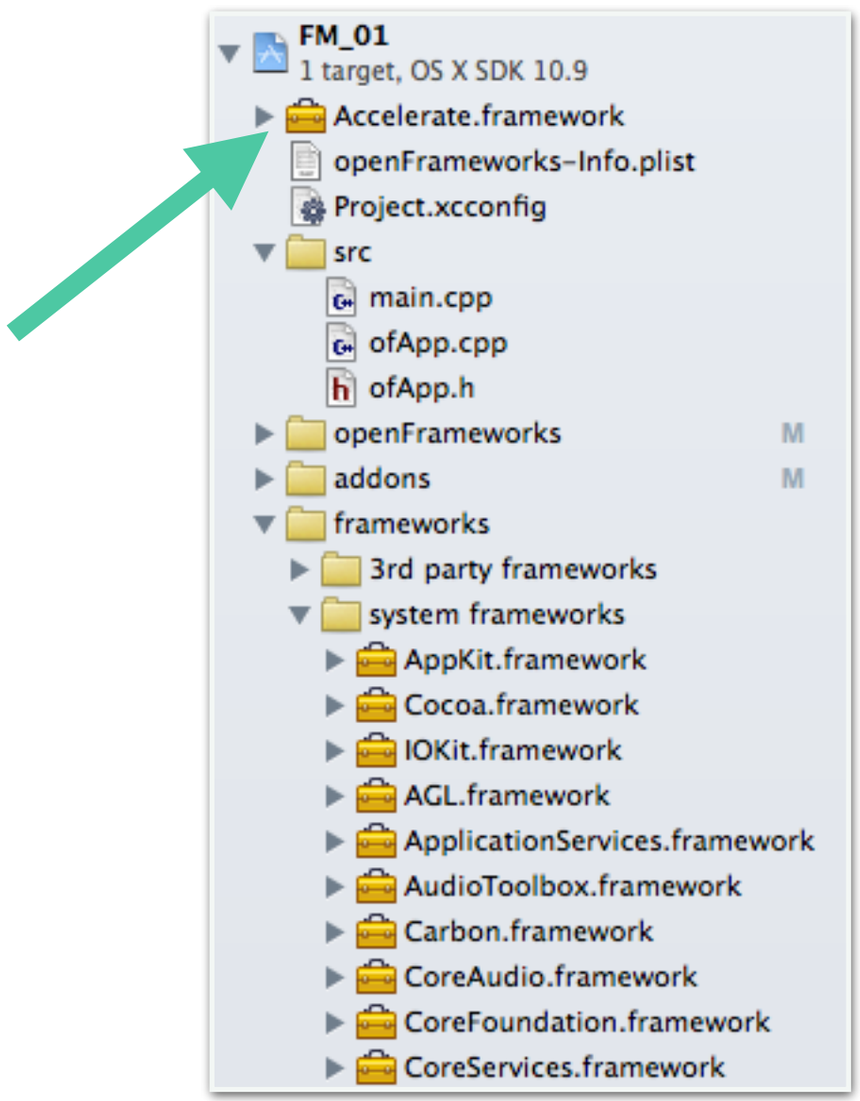
import Accelerate.framework

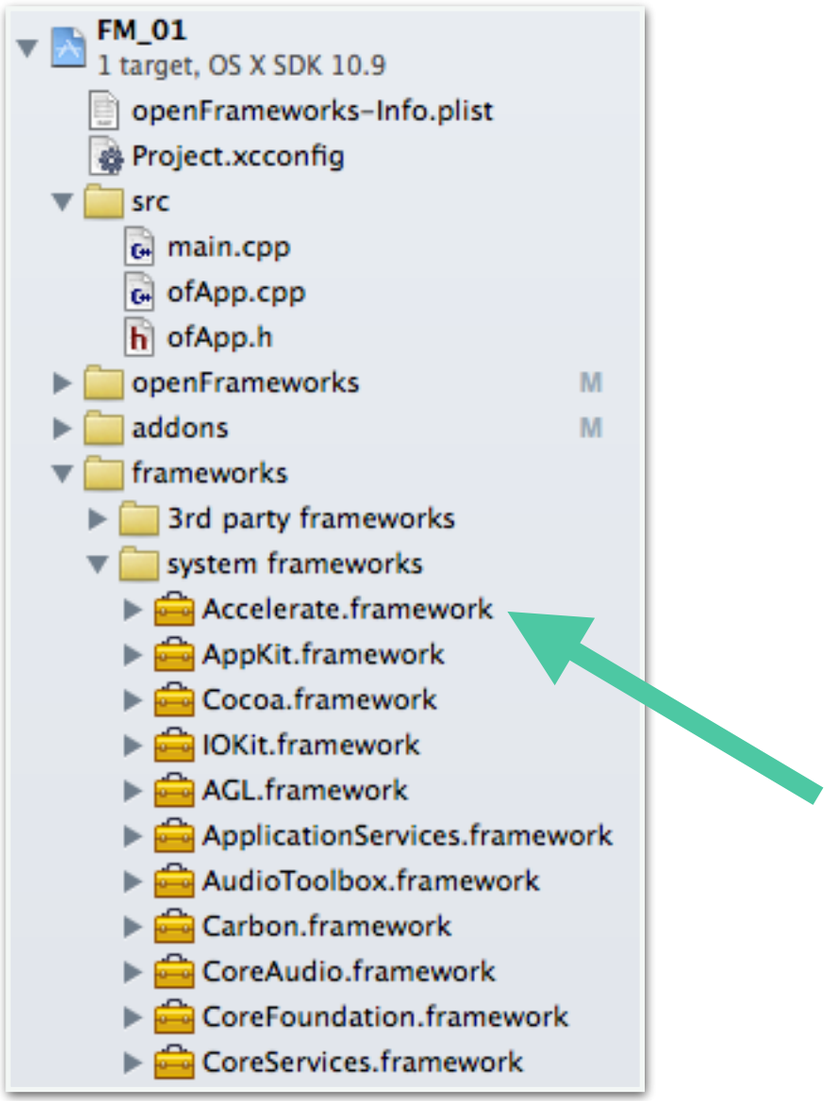


import Accelerate.framework



import Accelerate.framework





code basic

code

- .h

```
#include "ofxTonic.h"

using namespace Tonic;

ofxTonicSynth synth;
```

- .cpp

```
ofSoundStreamSetup(2, 0, this, 44100, 256, 4);

__SyntCode__

synth.setOutputGen( __output__ );


void ofApp::audioRequested(float* output, int bufferSize, int nChannels){
    synth.fillBufferOfFloats(output, bufferSize, nChannels);
}
```

simple Tone

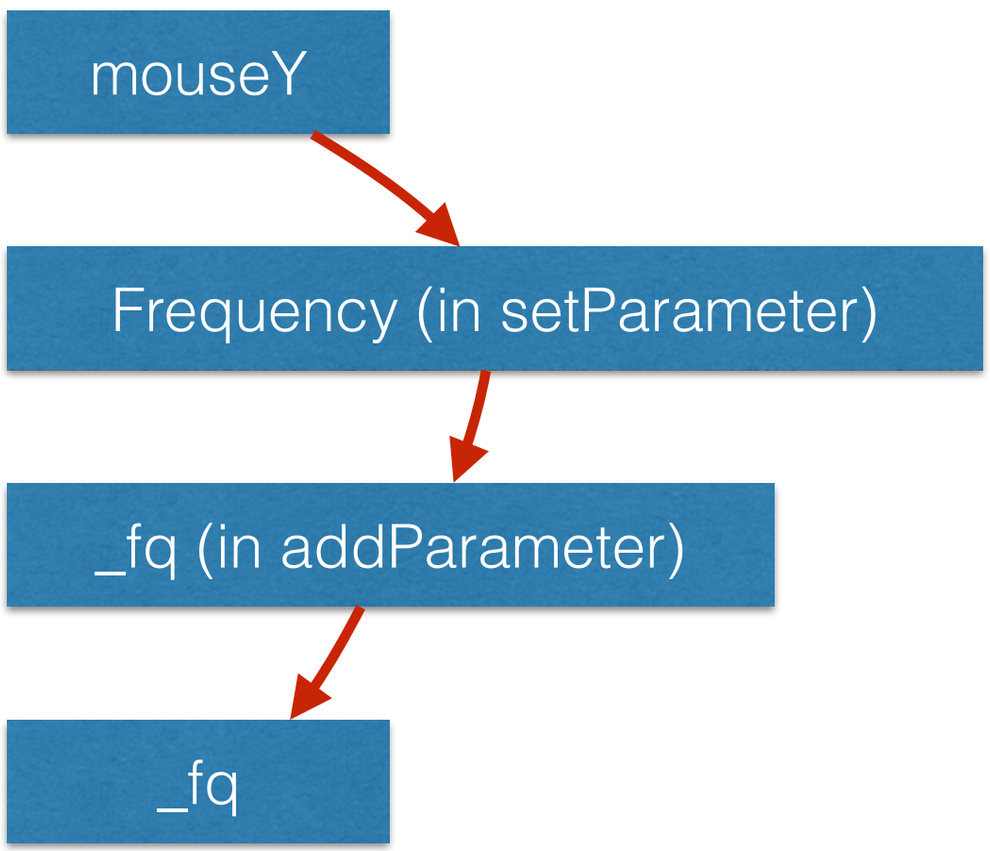
synth code

```
Generator sinTone = SineWave().freq(440);

synth.setOutputGen(sinTone);
```

simple Parameter

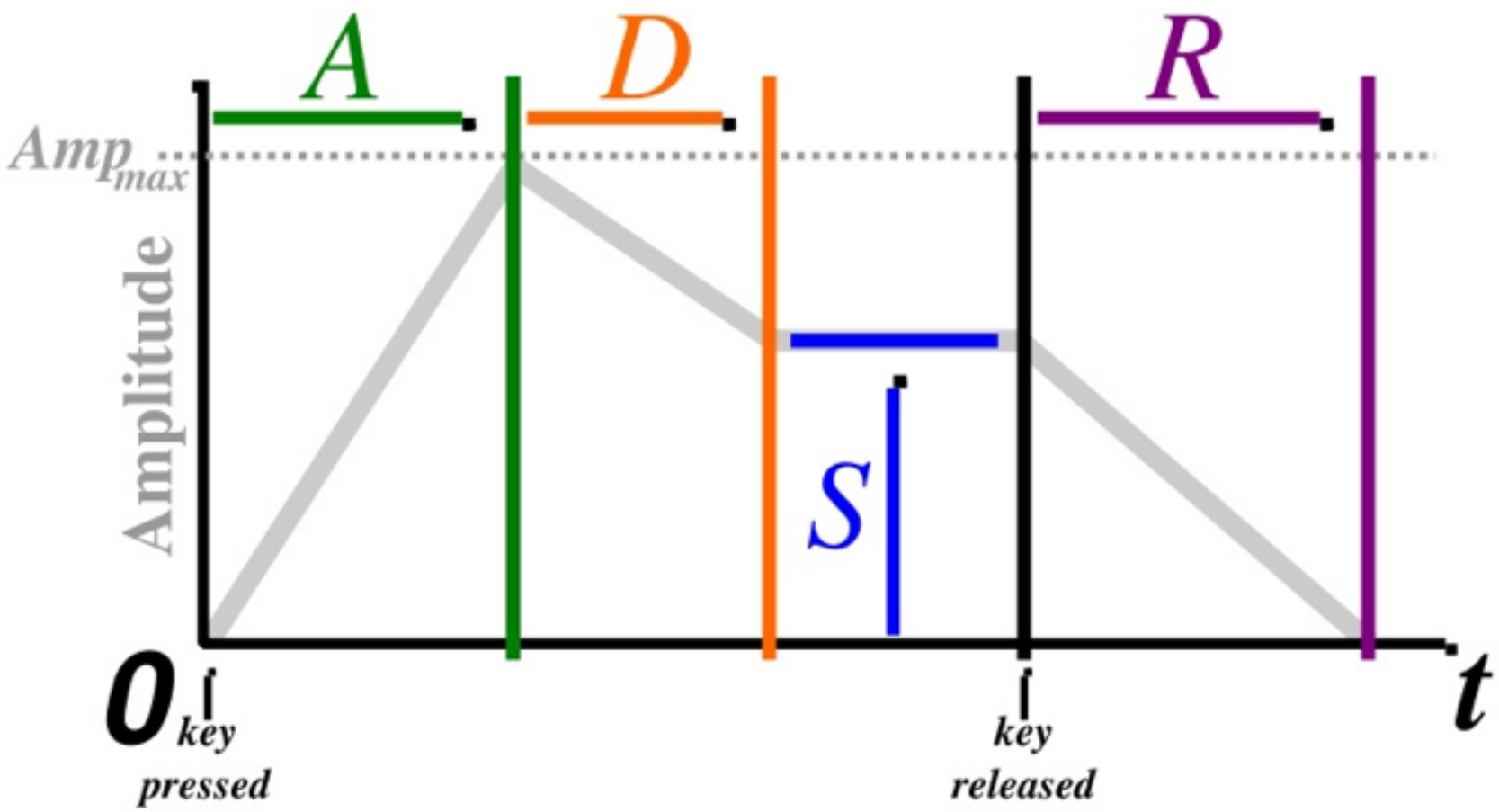
code

```
ControlParameter _fq = synth.addParameter(„Frequency");
Generator sinTone = SineWave().freq(_fq);

synth.setParameter("Frequency", mouseY);
```



simple Trigger

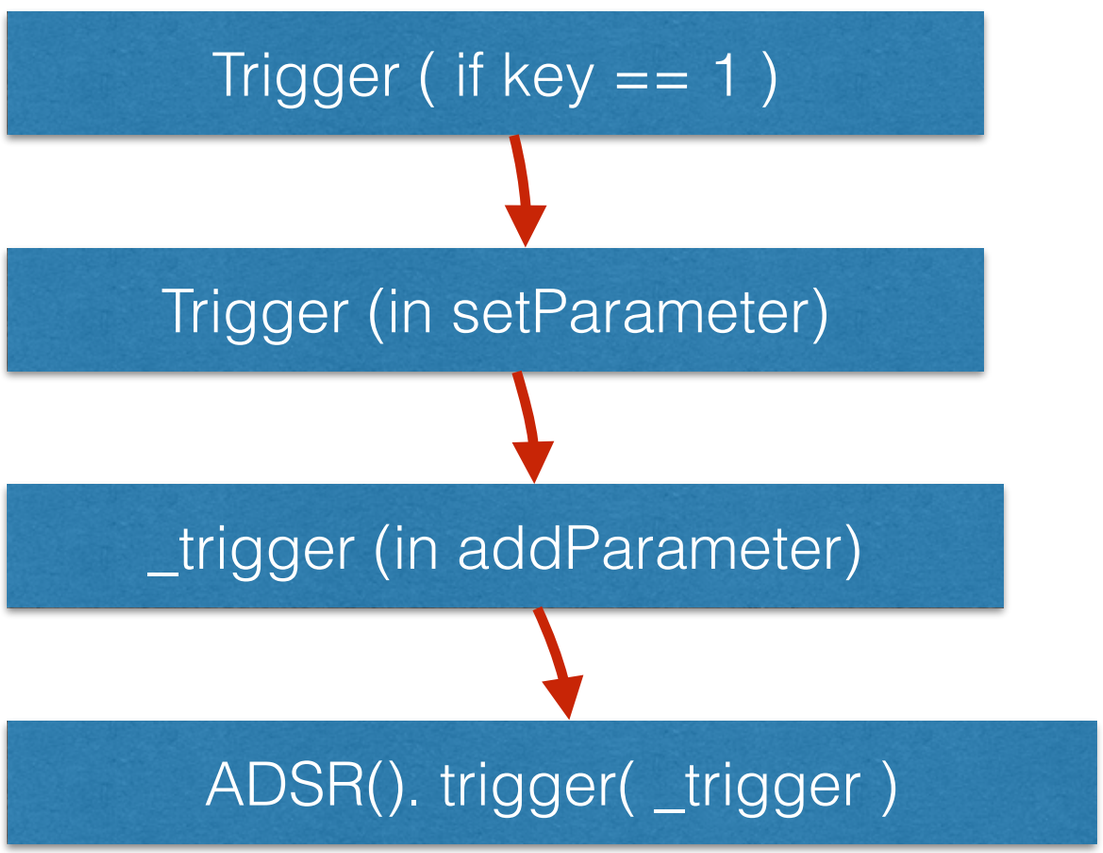
ADSR

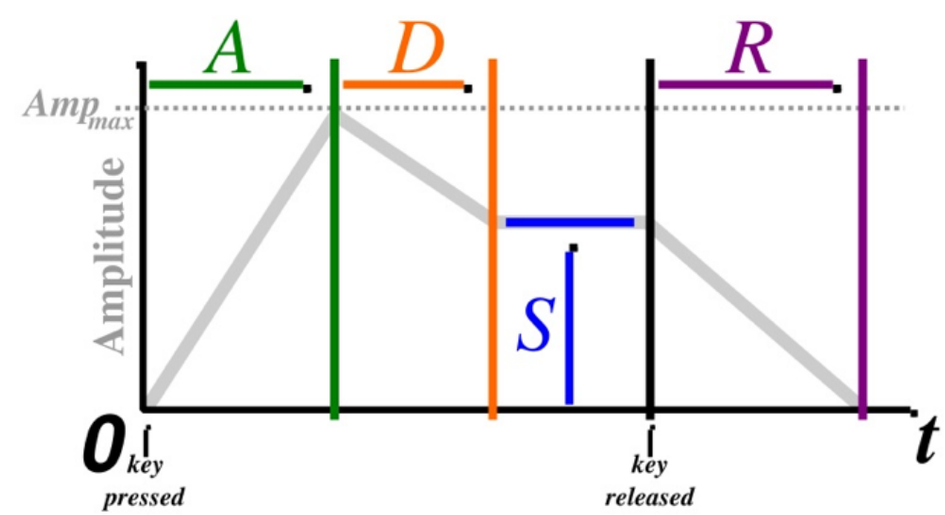


http://en.wikipedia.org/wiki/Synthesizer#ADSR_envelope

code

```
ControlParameter _trigger = synth.addParameter("Trigger");
Generator env = ADSR().attack(0).decay(1).sustain(0).release(0).trigger(_trigger);

synth.setParameter("Trigger", 1);
synth.setParameter("Frequency", 440);
```



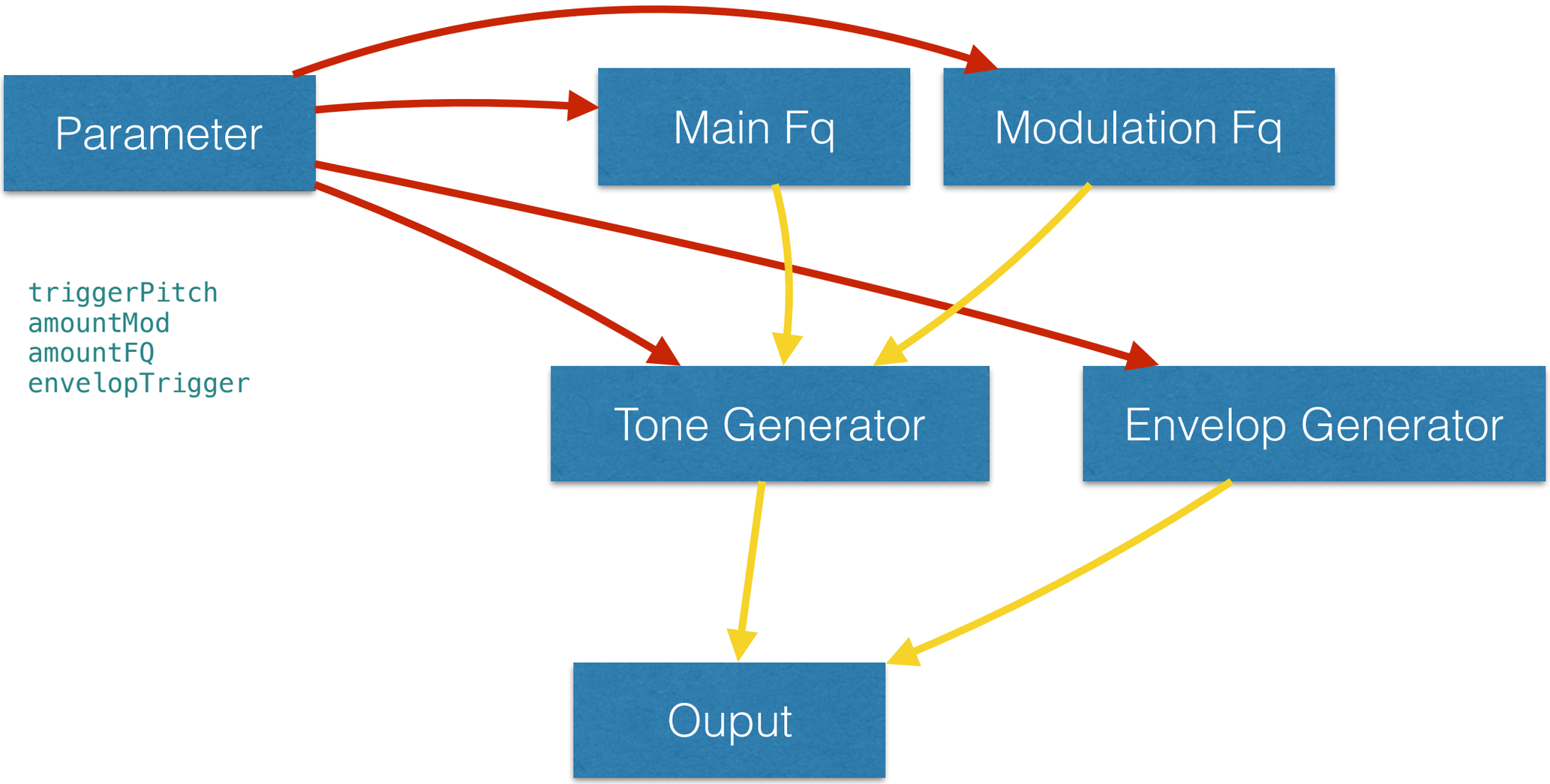


FM_01 Code

synth code

```
- in setup()

// Parameter
ControlParameter triggerPitch = synth.addParameter("triggerPitch");
ControlParameter amountMod = synth.addParameter("amountMod");
ControlParameter amountFQ = synth.addParameter("amountFQ");
ControlParameter envelopTrigger = synth.addParameter("trigger");

// Main Fq
Generator mainFq = ControlMidiToFreq().input(triggerPitch).smoothed();

// Modulation Fq
Generator rModFq     = mainFq * amountFQ;
Generator modulation = SineWave().freq( rModFq ) * rModFq * amountMod;

// Tone Generator
Generator tone = SineWave().freq(mainFq + modulation);

// Envelop Generator
Generator env = ADSR().attack(0.001).decay(0.5).sustain(0).release(0).trigger(envelopTrigger).legato(false);

// Output
synth.setOutputGen( tone * env * 0.75 );

ofAddListener(ofEvents().keyPressed, this, &ofApp::keyPressedOne);
```

synth diagram



code diagram

KeyPressed

```
synth.setParameter("trigger", 1);
synth.setParameter("triggerPitch", 48);
```

```
ControlParameter triggerPitch = synth.addParameter(„triggerPitch“);
ControlParameter envelopTrigger = synth.addParameter("trigger");
```

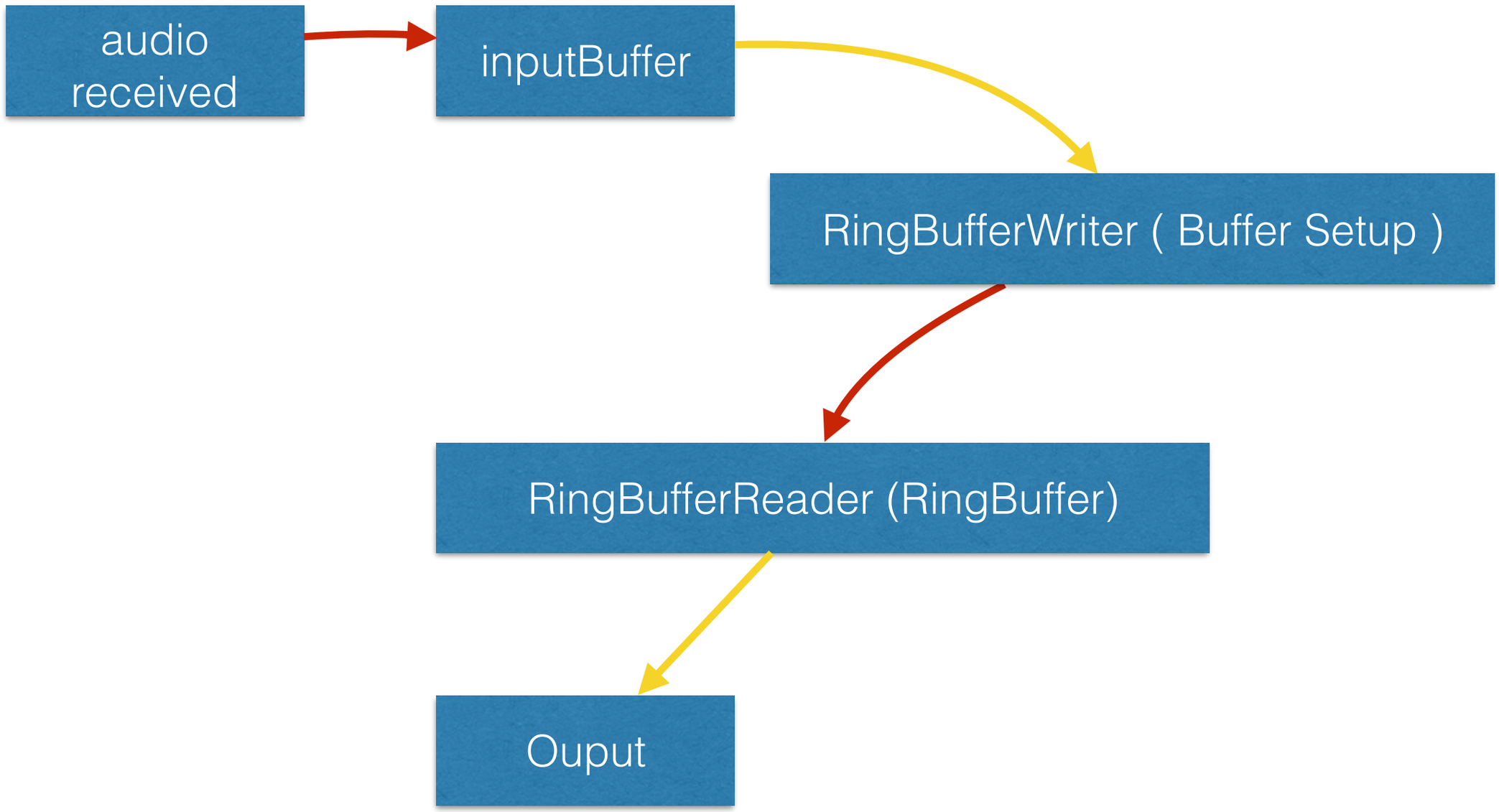
```
Generator mainFq = ControlMidiToFreq().input(triggerPitch).smoothed();
Generator env = ADSR().~~trigger(envelopTrigger).legato(false);
```

Input

code

```
-.h

void audioReceived(float* input, int bufferSize, int nChannels);
RingBufferWriter inputBuffer;


-.cpp

ofSoundStreamSetup(2, 2, this, 44100, 256, 4);
RingBuffer _inputS;
inputBuffer = RingBufferWriter("_inputS", 256, 4);
RingBufferReader inputReader = RingBufferReader().bufferName(„_inputS");

void ofApp::audioReceived(float* input, int bufferSize, int nChannels){
    inputBuffer.write(input, bufferSize, nChannels);
}
```

input diagram

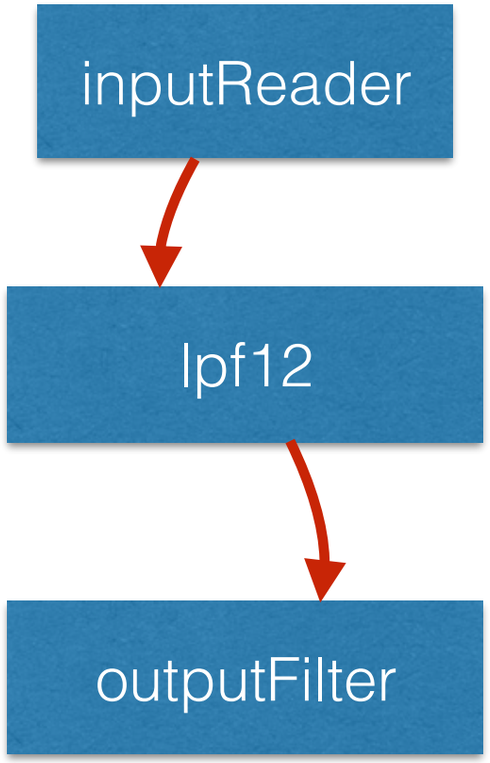


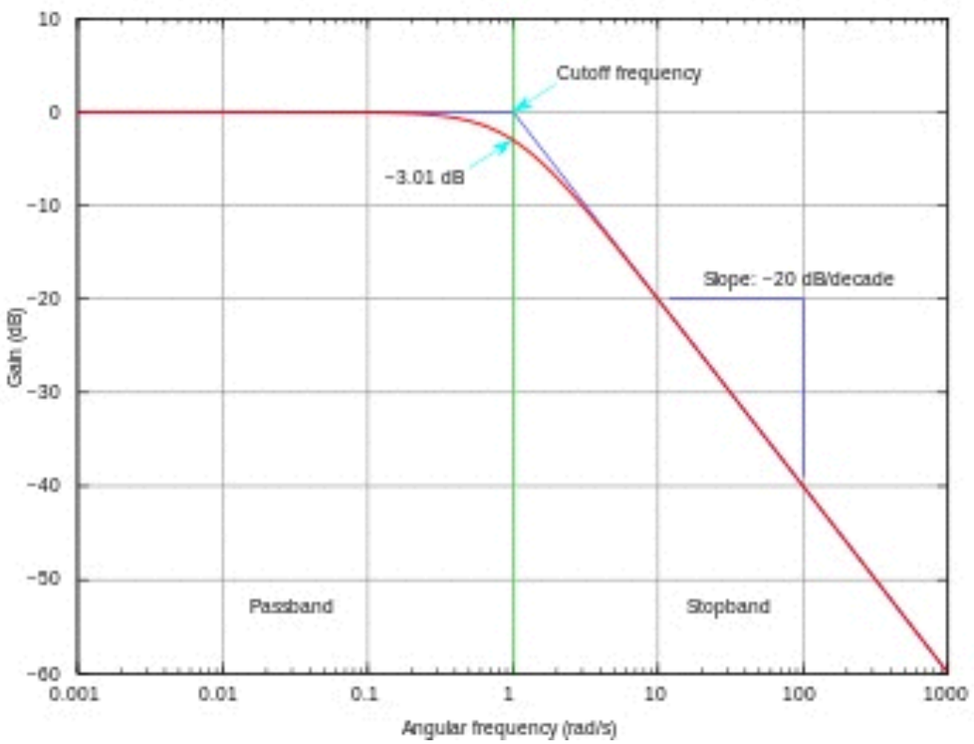
Filter

code

```
LPF12 lpf12 = LPF12().Q(10).cutoff(400);

Generator outputFilter = inputReader >> lpf12;
```

LPF
HPF
BPF



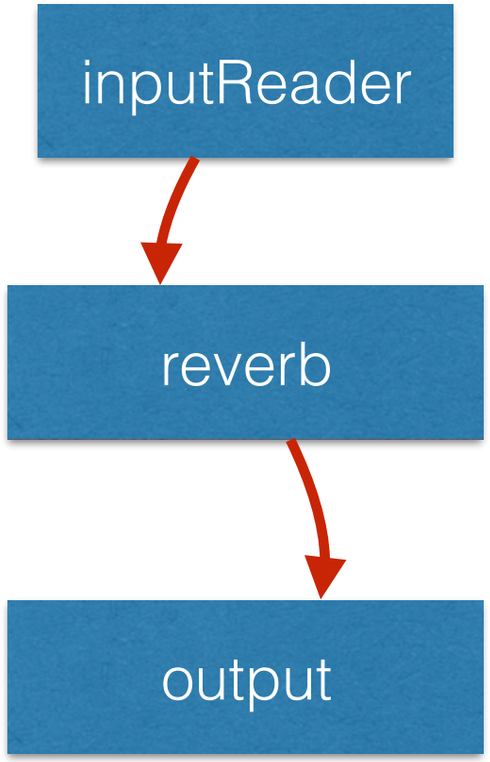


http://en.wikipedia.org/wiki/Low_pass_filter

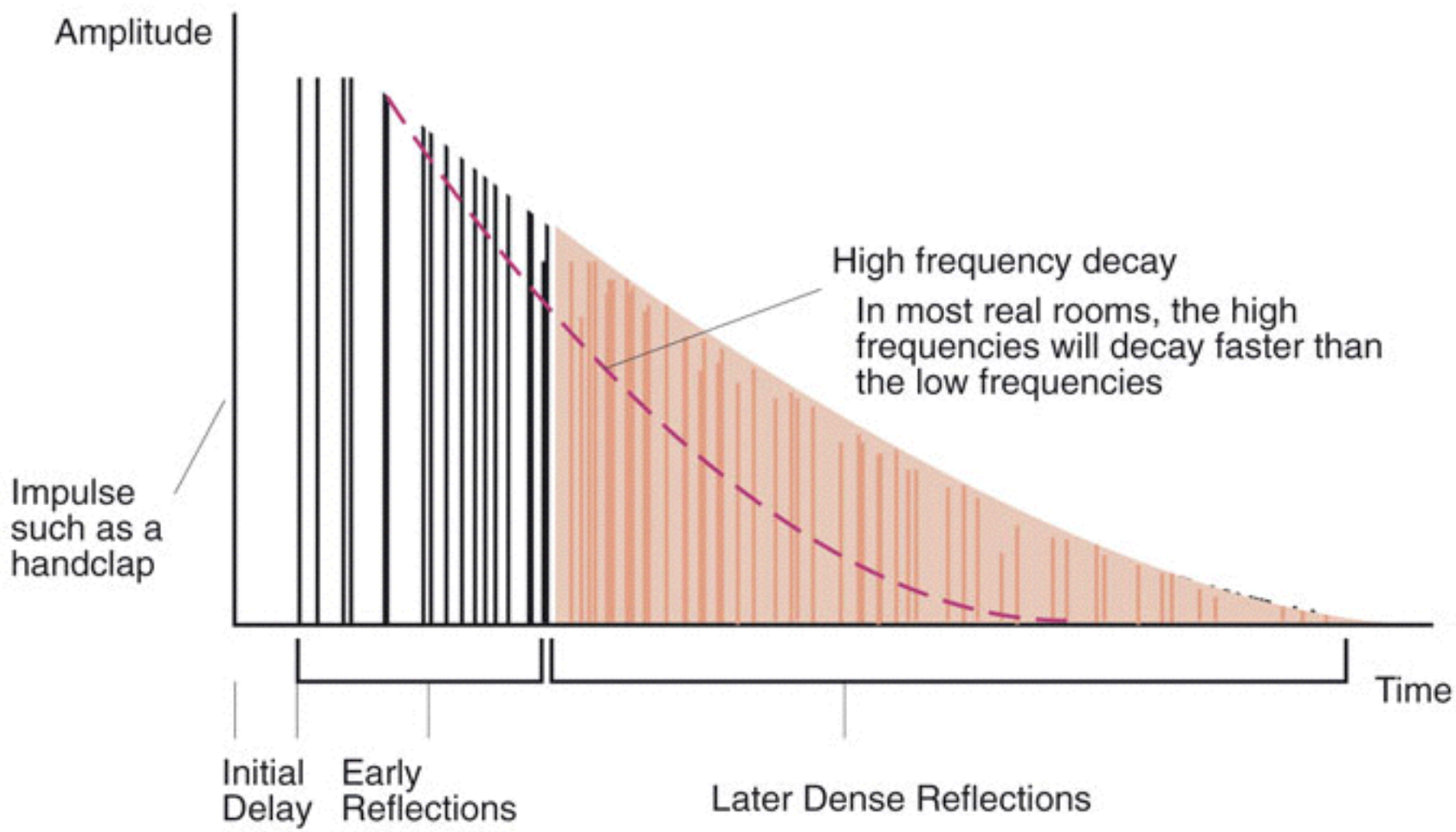
Reverb

code

```
ControlParameter __Parameter__ = synth.addParameter(„__NAME__“,__VALUE__);

Reverb reverb = Reverb()
.preDelayTime(preDelay)
.inputLPFCutoff(inputLPF)
.inputHPFCutoff(inputHPF)
.decayTime(time)
.decayLPFCutoff(lowDecay)
.decayHPFCutoff(hiDecay)
.stereoWidth(stereo)
.density(density)
.roomShape(shape)
.roomSize(size)
.dryLevel(ControlDbToLinear().input(dry))
.wetLevel(ControlDbToLinear().input(wet));

Generator output = inputReader >> reverb;
```



code



http://www.soundonsound.com/sos/jul08/articles/reverb1.htm