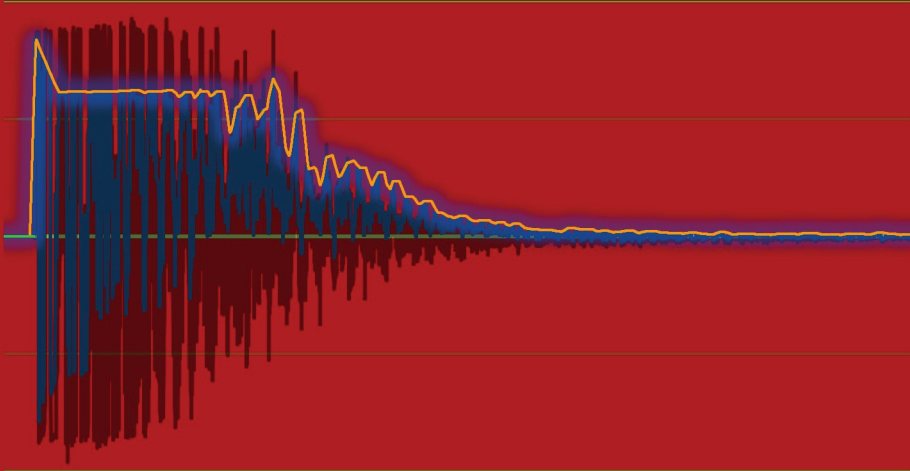


# Digital Signal Processing for Audio Applications

Volume 2 - Code  
Third Edition

Anton Kamenov



# **Digital Signal Processing for Audio Applications**

**Third Edition**  
**Volume 2 – Code**

Anton Kamenov

© 2017 Anton Kamenov. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the prior written permission of the author.

The author does not offer any warranties or representations and does not accept any liabilities with respect to the examples presented in this book.

June 2017

ISBN-13: 978-0-692-91381-9

**Foreword to the First Edition of Digital Signal Processing for Audio Applications**

*In the summer of 2003 we began designing multi-track recording and mixing software – Orinj at RecordingBlogs.com – a software application that will take digitally recorded audio tracks and will mix them into a complete song with all the needed audio production effects. Manipulating digital sound, as it turned out, was not easy. We had to find the answers of many questions, including what digital audio was, how we could mix audio tracks, how we could track the amplitude of digital sound so that we could apply compression, how we could track frequencies so that we could equalize, what a good model of artificial reverb would be, and many others. Bits of relevant information were available, albeit not always well organized and not always intuitive.*

*"Digital Signal Processing for Audio Applications" provides much of the needed information. It is a simple structured approach to understanding how digitally recorded sound can be manipulated. It presents and explains, and sometimes derives, the mathematical theory that the DSP user can employ in designing sound manipulating applications.*

*Although this book introduces much mathematics, we have designed it not for mathematicians, but for the engineers and hobbyists, who would be interested in the practical applications of DSP and not in its theoretical derivations. If properly explained, much of the practical DSP applications reduce to simple algebra. This said, we have included a sufficient amount of theory to provide an explanation of why DSP works the way it does. It is important for practitioners to have a good understanding of how DSP concepts come about. Much of the available DSP information has too much theory and not enough examples. Much of it has too many practical examples and not enough theoretical backing. We hope to have found the proper balance.*

*We hope you enjoy this book and make use of its definitions, explanations, and numerous examples.*

*The author and the administrators of [www.recordingblogs.com](http://www.recordingblogs.com)*

## **Foreword to The Code**

*Explaining the mathematics behind digital signal processing – DSP – is the task volume 1. It is a start, but there is more. It is not always straightforward to translate the mathematics into code. The purpose of volume 2 is just that. It translates the mathematical formulae in volume 1 into practical algorithms. It does so with actual DSP effects, including distortion, delay, chorus, equalizer, compressor, reverb, wah wah, and others.*

*Volume 1 of this book makes the argument that much of DSP can be reduced to simple algebraic and trigonometric manipulations. We hope that this volume shows that coding DSP is similarly not complex. In contemporary audio recording and mixing software, storing audio data, managing audio files, and designing an intuitive but functional user interface could be much more intricate than modifying the audio data themselves.*

*We hope you make use of this book and design some of your own DSP effects. They may just sound better than anyone else's. Audio production is as much an art, as it is science.*

*Many thanks to Mic of RecordingBlogs.com for providing access to the Orinj source code.*

*The author and the administrators of [www.recordingblogs.com](http://www.recordingblogs.com)*

## Table of Contents

Chapter 1. Introduction.....	12
1.1. Reading this book.....	12
Chapter 2. The WAVE file format.....	14
2.1. RIFF chunk.....	14
2.2. Wave chunks.....	14
2.3. Endianism.....	14
2.4. Word alignment.....	15
2.5. Format chunk.....	15
2.6. Data chunk.....	17
2.7. An example of an actual wave file.....	17
2.8. Other wave chunks.....	20
Chapter 3. The Orinj effect framework.....	21
3.1. oreffect.jar.....	21
3.2. EffectFont.java.....	22
3.3. EffectInterface.java.....	22
3.4. EffectPanelInterface.java.....	26
3.5. ReadInterface.java.....	27
3.6. Undo.java.....	28
3.7. UndoEvent.java.....	29
3.8. UndoListener.java.....	30
3.9. WriteInterface.java.....	30
Chapter 4. Distortion.....	32
4.1. Implementation of the distortion.....	32
4.2. Using audio buffers.....	38
4.3. Other sampling rates and resolutions.....	38
4.4. Error checking.....	40
4.5. Implementation of the distortion graphical user interface.....	41
4.6. Error checking in the effect panel interface.....	46
4.7. Undo.....	47
4.8. Implementation in effect.xml.....	48
4.9. Packaging.....	49
4.10. Obfuscating.....	49
Chapter 5. Testing the distortion.....	50
5.1. AudioBuffer.java.....	50
5.2. EffectDialog.java.....	52
5.3. EffectStore.java.....	54

---

5.4. EffectStoreItem.java .....	54
5.5. ExampleDelayTest.java .....	56
5.6. MainFrame.java .....	59
5.7. Mixer.java .....	65
5.8. SelectDialog.java.....	70
5.9. WaveFile.java .....	73
5.10. WaveFileDialog.java.....	77
5.11. Compiling ExampleDelayTest.....	81
Chapter 6. Delay .....	83
6.1. Implementation of the delay .....	83
6.2. Using past audio buffers .....	89
6.3. Signal polarity.....	90
6.4. Working with stereo data.....	90
6.5. Complex settings for the simple delay.....	93
6.6. Implementation of the delay graphical user interface.....	94
6.7. Creating presets in Orinj.....	100
Chapter 7. Echo.....	101
Chapter 8. Multitap delay .....	105
Chapter 9. Chorus .....	113
Chapter 10. Bass chorus .....	121
10.1. High pass filter .....	121
10.2. Implementation of the bass chorus.....	122
10.3. Delays in phase.....	128
Chapter 11. Equalizer.....	130
11.1. Low pass, high pass, and band pass filters .....	130
11.2. The equalizing filter .....	133
11.3. Implementation of the equalizer.....	136
11.4. Magnitude response of the equalizer .....	139
11.5. Phase response of the equalizer.....	140
Chapter 12. Noise gate.....	141
12.1. Hilbert transform .....	141
12.2. Implementation of the noise gate.....	142
Chapter 13. Compressor.....	147
13.1. Practical compression and expansion of dynamics.....	147
13.2. An example of drum compression .....	148
13.3. Implementation of the simple compressor .....	152
13.4. Forward-looking compressor.....	157
13.5. Average vs. peak compression .....	159

---

13.6. Side chained compressor.....	159
13.7. Multiband compressor.....	159
13.8. Multi-threshold compressor .....	160
13.9. Soft-knee compression.....	160
Chapter 14. Reverb.....	165
14.1. Implementation of the reverb .....	165
14.2. Properties of the digital reverb.....	175
Chapter 15. Wah wah.....	178
Chapter 16. Pitch shift .....	186
16.1. Actual frequencies.....	186
16.2. Overlap .....	186
16.3. Pitch shifting through changes in the phase .....	187
16.4. Windowing.....	188
16.5. Fast Fourier transform .....	188
16.6. Implementation of the pitch shift.....	188
16.7. Stretching or shrinking.....	195
Chapter 17. Mixing and recording.....	196
17.1. Mixing several tracks.....	196
17.2. Recording .....	199
Appendix A. Other wave chunks .....	204
A.1. Cue chunk.....	204
A.2. Fact chunk.....	205
A.3. Instrument chunk.....	206
A.4. List chunk.....	206
A.5. Playlist chunk .....	209
A.6. Sample chunk.....	210
A.7. Silent chunk.....	212
A.8. Wave list chunk.....	213



## Table of Figures

Figure 1. Structure of the format chunk in a wave file .....	15
Figure 2. Structure of the data chunk in a wave file .....	17
Figure 3. Order of samples in stereo WAVE data.....	19
Figure 4. Schema for a multitap delay .....	105
Figure 5. Magnitude response of high pass filters with three different lengths .....	122
Figure 6. Magnitude response of the Hilbert transform.....	142
Figure 7. An example snare hit.....	149
Figure 8. Amplitude envelope of an example snare hit .....	149
Figure 9. Applying compression on the amplitude envelope of the snare hit.....	150
Figure 10. Amplitude envelope of the snare hit before and after the compression .....	151
Figure 11. Original and compressed snare hit.....	151
Figure 12. Soft-knee compression.....	161
Figure 13. Magnitude response of the wah wah filter.....	178
Figure 14. 75 Hz from overlapping 50 Hz segments .....	187
Figure 15. Structure of the cue chunk in a wave file .....	204
Figure 16. Structure of a cue point in a cue chunk.....	204
Figure 17. Structure of a fact chunk in a wave file .....	205
Figure 18. Structure of an instrument chunk in a wave file .....	206
Figure 19. Structure of a list chunk in a wave file.....	207
Figure 20. Structure of a label or a note sub-chunk .....	207
Figure 21. Structure of a labeled text sub-chunk .....	207
Figure 22. Common info IDs .....	208
Figure 23. Structure of a playlist chunk in a wave file.....	209
Figure 24. Structure of a segment of a playlist chunk .....	210
Figure 25. Structure of a sample chunk in a wave file.....	210
Figure 26. Structure of a sample loop.....	212
Figure 27. Structure of a silent chunk in a wave file.....	212
Figure 28. Structure of a wave list chunk in a wave file.....	213

## Table of Code Samples

Code 1. An example start of a WAVE file.....	14
Code 2. Contents of an example wave file.....	17
Code 3. EffectFont.java.....	22
Code 4. EffectInterface.java.....	22
Code 5. EffectPanelInterface.java.....	26
Code 6. ReadInterface.java.....	27
Code 7. Undo.java.....	28
Code 8. UndoEvent.java.....	29
Code 9. UndoListener.java.....	30
Code 10. WriteInterface.java.....	30
Code 11. Distortion.....	32
Code 12. Bytes to 16-bit samples and 16-bit samples to bytes.....	39
Code 13. Bytes to 8-bit samples and 8-bit samples to bytes.....	39
Code 14. Bytes to 24-bit samples and 24-bit samples to bytes.....	39
Code 15. Bytes to 32-bit samples and 32-bit samples to bytes.....	40
Code 16. Example error checking in the simple delay.....	40
Code 17. Distortion graphical user interface.....	41
Code 18. Error checks in the graphical user interface – focusLost.....	46
Code 19. Error checks in the graphical user interface – actionPerformed.....	47
Code 20. Undo in the distortion when setting the threshold.....	47
Code 21. effect.xml file for the distortion.....	48
Code 22. Packaging the distortion.....	49
Code 23. AudioBuffer.java.....	50
Code 24. EffectDialog.java.....	52
Code 25. EffectStore.java.....	54
Code 26. EffectStoreItem.java.....	54
Code 27. ExampleDelayTest.java.....	57
Code 28. MainFrame.java.....	60
Code 29. Mixer.java.....	65
Code 30. SelectDialog.java.....	70
Code 31. WaveFile.java.....	73
Code 32. WaveFileDialog.java.....	77
Code 33. Compiling ExampleDelayTest.....	81
Code 34. Manifest for ExampleDelayTest.....	82
Code 35. Delay.....	83
Code 36. Storing audio data.....	89
Code 37. Finding the right audio data.....	89
Code 38. Removing unused past audio data.....	90
Code 39. Incrementing buffer indices of past audio data.....	90

---

Code 40. The delay apply function for stereo audio .....	90
Code 41. Delay graphical user interface .....	94
Code 42. Echo constructor .....	102
Code 43. Echo at the beginning of playback.....	102
Code 44. Echo .....	102
Code 45. A multitap delay tap .....	106
Code 46. Computing the impulses of a multitap delay .....	107
Code 47. Multitap delay constructor.....	109
Code 48. Multitap delay at the beginning of playback .....	110
Code 49. Multitap delay .....	110
Code 50. Chorus constructor.....	115
Code 51. Chorus at the beginning of playback .....	116
Code 52. Chorus .....	117
Code 53. High pass filter .....	121
Code 54. Bass chorus constructor.....	123
Code 55. Bass chorus hasData .....	123
Code 56. Bass chorus at the beginning of playback .....	123
Code 57. Bass chorus .....	124
Code 58. Bass chorus delays unadjusted for the phase delay.....	128
Code 59. Bass chorus delays adjusted for the phase delay .....	129
Code 60. Low pass filter .....	130
Code 61. Blackman window .....	131
Code 62. Bartlett-Hann window .....	131
Code 63. High pass filter .....	131
Code 64. Band stop filter.....	132
Code 65. Band pass filter.....	132
Code 66. An equalizer band.....	134
Code 67. Equalizer bands.....	135
Code 68. Equalizer band filters .....	135
Code 69. Total equalizer filter .....	136
Code 70. Equalizer constructor.....	136
Code 71. Equalizer at the beginning of playback .....	137
Code 72. Equalizer .....	137
Code 73. Magnitude response of the equalizer .....	139
Code 74. Dry and wet mix in the equalizer .....	140
Code 75. Hilbert transform.....	141
Code 76. Noise gate constructor.....	143
Code 77. Noise gate at the beginning of playback.....	143
Code 78. Noise gate .....	143
Code 79. Compressor constructor.....	153
Code 80. Compressor at the beginning of playback.....	153

---

Code 81. Compressor.....	153
Code 82. Forward looking compression index .....	158
Code 83. Past buffers in forward looking compressors.....	158
Code 84. Applying gain in a forward looking compressor .....	159
Code 85. Soft-knee compression.....	161
Code 86. Reverb constructor .....	167
Code 87. Reverb at the beginning of playback.....	167
Code 88. Reverb.....	167
Code 89. Wah wah constructor .....	180
Code 90. Wah wah at the beginning of playback.....	180
Code 91. Wah wah.....	181
Code 92. Pitch shift constructor.....	189
Code 93. Pitch shift at the beginnning of playback.....	190
Code 94. Pitch shift.....	191
Code 95. Mixing with getData .....	196
Code 96. Start recording.....	199
Code 97. Record .....	201

## Index

### A

addUndoListener, 18  
 all pass filter, 153  
 allowsDryWetMix, 13  
 allowsSideChaining, 13  
 apply, 12  
   bass chorus, 112  
   chorus, 106  
   compressor, 142  
   delay, 74  
   delay (stereo), 79  
   distortion, 23  
   echo, 91  
   equalizer, 125  
   multitap delay, 99  
   noise gate, 131  
   pitch shift, 179  
   reverb, 155  
   wah wah, 169  
 audio buffers, 27, 78  
 AudioBuffer, 39

### B

band pass filter, 120, 166  
 band stop filter, 120  
 Bartlett-Hann window, 119  
 bass chorus, 109  
   apply, 112  
   constructor, 111  
   hasData, 111  
   phase, 116  
   startPlay, 111  
 Blackman window, 118  
 block align, 5  
 bounce, 82, 89

### C

chamber reverb unit, 163  
 chorus, 83, 102  
   apply, 106  
   bass. *See* bass chorus  
   constructor, 104

  startPlay, 105  
 clip, 21  
 comb filter, 72  
   feedback, 90  
   feedforward, 72, 153  
 compression code, 4  
 compressor, 135  
   apply, 142  
   average vs. peak, 147  
   constructor, 141  
   forward-looking, 145  
   multiband, 147  
   multi-threshold, 148  
   side chained, 147  
   soft-knee, 148  
   startPlay, 141  
 constructor  
   bass chorus, 111  
   chorus, 104  
   compressor, 141  
   delay, 72  
   distortion, 21  
   echo, 91  
   equalizer, 124  
   multitap delay, 98  
   noise gate, 131  
   pitch shift, 177  
   reverb, 155  
   wah wah, 168  
 crossfade, 167, 174

### D

data chunk, 6  
 delay, 72  
   apply, 74, 79  
   bounce, 82, 89  
   constructor, 72  
   graphical user interface, 83  
   multitap. *See* multitap delay  
   settings, 82  
   slapback. *See* slapback delay  
   spatial positioning, 82  
   startPlay, 73  
   sweep, 102  
   updateData, 88

DFT. *See* discrete Fourier transform  
 discrete Fourier transform, 174  
 distortion, 21  
     apply, 23  
     constructor, 21  
     graphical user interface, 29  
     startPlay, 23  
     updateData, 34

**E**

echo, 82, 90  
     apply, 91  
     constructor, 91  
     startPlay, 91  
 effect.xml, 37  
 EffectDialog, 41  
 EffectFont, 11  
 EffectInterface, 11  
 EffectPanelInterface, 15  
 EffectStore, 43  
 EffectStoreItem, 43  
 endianism, 3  
 equalizer, 118, 153  
     apply, 125  
     Band, 122  
     computeFilter, 123  
     constructor, 124  
     startPlay, 125  
 error checking, 29, 34  
 ExampleDelayTest, 45  
     compile, 70  
     manifest, 70

**F**

fast Fourier transform, 176  
 feedback comb filter, 90  
 feedforward comb filter, 72, 153  
 FFT. *See* fast Fourier transform  
 fireUndoEvent, 18  
 format chunk, 4  
 Fourier transform, 174

**G**

gate. *See* noise gate  
 getData, 65, 184  
 getUndoMessage, 19  
 graphical user interface

delay, 83  
 distortion, 29

**H**

Hann window, 176  
 hard clip, 21  
 hasData, 14  
     bass chorus, 111  
 high pass filter, 109, 119  
 Hilbert transform, 129

**L**

loadEffects, 49  
 low pass filter, 118

**M**

magnitude response, 127  
 MainFrame, 48  
 Mixer, 53  
 mixing, 184  
 multitap delay, 94  
     apply, 99  
     constructor, 98  
     startPlay, 99

**N**

noise gate, 129  
     apply, 131  
     constructor, 131  
     startPlay, 131

**O**

obfuscating, 38  
 openRead, 63  
 oreffect.jar, 10  
 Orinj effect framework, 10

**P**

packaging, 37  
 PCM. *See* pulse code modulation  
 phase, 79, 93, 116, 128  
 pitch shift, 174  
     apply, 179  
     constructor, 177

startPlay, 178  
plate reverb unit, 164  
presets, 89  
pulse code modulation, 4

## R

ReadInterface, 16  
readObject, 14  
recording, 187  
    and simultaneous playback, 189  
removeUndoListener, 18  
reverb, 153  
    apply, 155  
    chamber reverb unit, 163  
    constructor, 155  
    plate reverb unit, 164  
    spring reverb unit, 164  
    startPlay, 155  
RIFF chunk, 3

## S

sampling rate, 5, 27  
sampling resolution, 5, 27  
SelectDialog, 59  
serialization, 14, 16  
setEqual, 15  
setLanguage, 15  
shelving filter, 166  
significant bits per sample. *See* sampling resolution  
slapback delay, 82  
soft clip, 21  
spatial positioning, 82  
spring reverb unit, 164  
startPlay, 14, 64  
    bass chorus, 111  
    chorus, 105  
    compressor, 141  
    delay, 73  
    distortion, 23  
    echo, 91  
    equalizer, 125  
    multitap delay, 99  
    noise gate, 131  
    pitch shift, 178  
    recording, 187  
    reverb, 155  
    wah wah, 168

WaveFile, 64  
stereo data, 79  
stopPlay, 14  
stretching / shrinking, 183  
sweep, 102

## T

tapped delay line, 94  
testing, 39  
transducer, 164

## U

undo, 36  
Undo, 17  
UndoEvent, 18  
UndoListener, 19  
undoStorageRequired, 19  
updateData, 16  
    delay, 88  
    distortion, 34

## W

wah wah, 166  
    apply, 169  
    constructor, 168  
    startPlay, 168  
WAVE file format, 3  
    associated data list, 195  
    block align, 5  
    compression code, 4  
    cue chunk, 191  
    cue point, 192  
    Data chunk, 6  
    endianism, 3  
    fact chunk, 193  
    format chunk, 4  
    info, 196  
    instrument chunk, 194  
    label sub-chunk, 195  
    labeled text sub-chunk, 195  
    list chunk, 194  
    note sub-chunk, 195  
    playlist chunk, 197  
    playlist segment, 197  
    RIFF chunk, 3  
    sample chunk, 198

sample loop, 199  
sampling rate, 5  
sampling resolution, 5  
silent chunk, 200  
wave list chunk, 201  
word alignment, 4  
WaveFile, 61  
WaveFileDialog, 66  
window  
    Bartlett-Hann, 119  
    Blackman, 118  
    Hann, 176  
word alignment, 4  
WriteInterface, 19  
writeObject, 14