

**Designing Software Synthesizers in C++
for RackAFX, VST3 and Audio Units
Will Pirkle
Final 6.1.14**

Table of Contents

- 1 Synthesizer Fundamentals**
 - 1.1 Synth Components
 - 1.2 Oscillators
 - 1.3 Filters
 - 1.4 Amplifiers
 - 1.5 Envelope Generators
 - 1.6 Basic Synth Architecture
 - 1.7 Fundamental Goals of Synth Patch Design
 - 1.8 Audio Data Formats for Plug-Ins
 - 1.9 Signal Processing Review
 - 1.10 Continuous Signals
 - 1.11 Discretized Signals
 - 1.12 The LaPlace and z-Transforms
 - 1.13 Aliasing

- 2 Writing Plug-Ins**
 - 2.1 Dynamic Link Libraries
 - 2.2 C and C++ Style DLLs
 - 2.3 The Application Programming Interface (API)
 - 2.4 API Function Comparisons
 - 2.5 The RackAFX™ Philosophy and API
 - 2.6 Writing RackAFX Plug-Ins
 - 2.7 Setting Up RackAFX
 - 2.8 Designing the User Interface in Prototype Phase
 - 2.9 Setting Up Continuous Controls
 - 2.10 Setting Up Indexed Controls: Radio Buttons
 - 2.11 Setting Up Indexed Controls: Sliders
 - 2.12 Placing Controls Inside the LCD Control
 - 2.13 Using the RackAFX GUI Designer
 - 2.14 The RackAFX Plug-In Object
 - 2.15 Writing VST3 Plug-Ins
 - 2.16 VST3: Processor and Controller
 - 2.17 The Common Object Model: COM
 - 2.18 VST3 Synth Plug-In Architecture
 - 2.19 Processor and Controller Declarations
 - 2.20 The Class Factory
 - 2.21 VST3 Conventions

- 2.22 Implementing a VST3 Plug-In: Controller
- 2.23 VST3 Controller Initialization
- 2.24 VST3 Controller Serialization
- 2.25 VST3 Controller MIDI Mapping
- 2.26 Implementing a VST3 Plug-In: Processor
- 2.27 VST3 Processor Initialization
- 2.28 VST3 Processor Serialization
- 2.29 VST3 Note Events, Control Changes and Rendering
- 2.30 Implementing a VST3 Plug-In: GUI Design
- 2.31 Using the VSTGUI® Drag and Drop Interface
- 2.32 VSTGUI Design Example
- 2.33 Implementing a VST3 Plug-In: Debugging
- 2.34 Writing Audio Unit (AU) Plug-Ins
- 2.35 AU XCode Projects
- 2.36 The Info.plist File
- 2.37 Managing AU Parameters
- 2.38: AU Conventions
- 2.39 AUSynth and AUInstrument Base Class
- 2.40 Implementing the AUSynth Object
- 2.41 Implementing the Cocoa View Objects
- 2.42 WPRotaryKnob
- 2.43 WPOptionMenuGroup
- 2.44 The View Factory and CocoaSynthView.plist
- 2.45 The View Event Listeners
- 2.46 The View Interface and Initialization
- 2.47 Implementing an AU Plug-In: Debugging

3 MIDI

- 3.1 MIDI Messages
- 3.2 Channel Voice Messages
- 3.3 Project: NanoSynth
- 3.4 NanoSynth: RackAFX
- 3.5 RackAFX Status Window
- 3.6 NanoSynth: VST3
- 3.7 MIDI Events in VST3
- 3.8 MIDI Controllers in VST3
- 3.9 NanoSynth: AU

4 Analog and Digital Signal Processing

- 4.1 Analog and Digital Building Blocks
- 4.2 Analog and Digital Transfer Functions

- 4.3 Digital Delay
- 4.4 Digital Differentiation
- 4.5 Analog and Digital Integration
- 4.6 The Bilinear z-Transform
- 4.7 Virtual Analog Filters
- 4.8 Analog Block Diagrams
- 4.9 1st Order VA Lowpass Filter
- 4.10 1st Order VA Highpass Filter
- 4.11 2nd Order VA Filters
- 4.12 Series and Parallel VA Filters
- 4.13 Resolving Delay-Less Loops: Modified Härmä Method
- 4.14 Resolving Delay-Less Loops: Zavalishin's Method
- 4.15 Analog Signal Flow Graphs
- 4.16 Wave Shaping
- 4.17 Challenges

5 Synthesizer Oscillator Design

- 5.1 Trivial Oscillator Algorithms
- 5.2 Trivial Sawtooth Oscillator
- 5.3 Trivial Square Wave Oscillator
- 5.4 Trivial Triangle Wave Oscillator
- 5.5 Quasi Bandlimited Oscillator Algorithms
- 5.6 Bandlimited Impulse Trains (BLIT)
- 5.7 Bandlimited Step Functions (BLEP)
- 5.8 Wider Window BLEP
- 5.9 Polynomial BLEP (PolyBLEP)
- 5.10 Coding the BLEP and PolyBLEP Algorithms
- 5.11 Wave Shaped BLEP Sawtooth Oscillator
- 5.12 BLEP Square Wave Oscillator
- 5.13 Differentiated Parabolic Waveform Oscillators
- 5.14 DPW Triangle Wave Oscillator
- 5.15 Other Non-Sinusoidal Oscillators
- 5.16 White Noise Oscillator
- 5.17 Pseudo Random Noise (PN Sequence) Oscillator
- 5.18 Random Sample and Hold Oscillator
- 5.19 Exponential Decay Oscillator
- 5.20 Wavetable Oscillators
- 5.21 Bandlimited Wavetable Oscillators
- 5.22 Sinusoidal Oscillators by Approximation
- 5.23 Pitched Oscillator Calculation
- 5.24 Sources of Modulation
- 5.25 Pitched Oscillator Starting Phase Consideration
- 5.26 LFO Features

- 5.27 Designing the Oscillator Objects
- 5.28 The COscillator Base Class
- 5.29 NanoSynth: Oscillators
- 5.30 NanoSynth Oscillators: RackAFX/VST3/AU
- 5.31 The CLFO Object
- 5.32 The CQBLimitedOscillator Object
- 5.33 The CWTOscillator Object
- 5.34 NanoSynth Oscillators: Processing Code
- 5.35 NanoSynth Oscillators: RackAFX
- 5.36 NanoSynth Oscillators: VST3
- 5.37 NanoSynth Oscillators: AU
- 5.38 NanoSynth: Wavetables
- 5.39 Challenges

6 Envelope Generators and Controlled Amplifiers

- 6.1 Analog Envelope Generators
- 6.2 Triggers, Gates and Reset to Zero
- 6.3 Note-On Modes
- 6.4 Note-Off Modes
- 6.5 Digital Envelope Generators
- 6.6 Envelope Generator Variations
- 6.7 Iterative Generation of the Exponential Function
- 6.8 Biased Envelope Generator
- 6.9 Envelope Generator Intensity Controls and Connections
- 6.10 Envelope Generator Implementation
- 6.11 The CEnvelopeGenerator Object
- 6.12 Digitally Controlled Amplifier (DCA)
- 6.13 The CDCA Object
- 6.14 NanoSynth: EG/DCA
- 6.15 NanoSynth EG/DCA Audio Processing
- 6.16 NanoSynth EG/DCA: RackAFX
- 6.17 NanoSynth EG/DCA: VST3
- 6.18 NanoSynth EG/DCA: AU
- 6.19 Challenges

7 Synthesizer Filter Design

- 7.1 Virtual Analog Filters
- 7.2 Oberheim SEM® State Variable Filter Model
- 7.3 Korg35 Sallen-Key Filter Model
- 7.4 Korg35 LPF Model
- 7.5 Korg35 HPF Model
- 7.6 Moog Ladder Filter Model
- 7.7 Moog Ladder Filter Gain Compensation

- 7.8 Oberheim XPander® Variations
- 7.9 Nonlinear Moog Ladder Filter Models
- 7.10 The 2nd Order Moog Half-Ladder Filter
- 7.11 Diode Ladder Filter Model
- 7.12 Designing the Filter Objects
- 7.13 The CFilter Base Class
- 7.14 The CVAOnePoleFilter Object
- 7.15 The CSEM Filter Object
- 7.16 The CKThreeFiveFilter Object
- 7.17 The CMoogLadderFilter Object
- 7.18 The CDiodeLadderFilter Object
- 7.19 Filter Key Tracking Modulation
- 7.20 NanoSynth: Filters
- 7.21 NanoSynth: Filters Audio Rendering
- 7.22 NanoSynth Filters: RackAFX
- 7.23 NanoSynth Filters: VST3
- 7.24 NanoSynth Filters: AU
- 7.25 Build and Test
- 7.26 Challenges

8 Modulation Matrix, Polyphony and Global Parametrization

- 8.1 Modulation Routings
- 8.2 The Modulation Matrix Object
- 8.3 Using and Programming the Modulation Matrix
- 8.4 NanoSynth MM Part 1: RackAFX
- 8.5 NanoSynth MM Part 1: VST3
- 8.6 NanoSynth MM Part 1: AU
- 8.7 More MIDI Modulation Routings
- 8.8 NanoSynth MM Part 2: RackAFX
- 8.9 NanoSynth MM Part 2: VST3
- 8.10 NanoSynth MM Part 2: AU
- 8.11 User Controlled Modulation Routings
- 8.12 NanoSynth MM Part 3: RackAFX
- 8.13 NanoSynth MM Part 3: VST3
- 8.14 NanoSynth MM Part 3: AU
- 8.15 Polyphony Part 1
- 8.16 Dynamic Voice Allocation
- 8.17 Poly NanoSynth: RackAFX
- 8.18 Poly NanoSynth: VST3
- 8.19 Poly NanoSynth: AU
- 8.20 Global Parameterization
- 8.21 Final NanoSynth: RackAFX

- 8.22 Final NanoSynth: VST3
- 8.23 Final NanoSynth: AU

9 MiniSynth: Analog Modeling Synth

- 9.1 Voice Architectures and CVoice
- 9.2 CVoice Initialization
- 9.3 Portamento
- 9.4 MiniSynth Specifications
- 9.5 CMiniSynthVoice Object
- 9.6 Polyphony Part 2
- 9.7 MiniSynth Files
- 9.8 MiniSynth: RackAFX
- 9.9 MiniSynth: VST3
- 9.10 MiniSynth: AU
- 9.11 Challenges

10 DigiSynth: Sample Playback Synthesizer

- 10.1 Audio Samples
- 10.2 WAV Files
- 10.3 Multi-Samples
- 10.4 Splits, Layers and Note Regions
- 10.5 The CWaveData Object
- 10.6 The CSampleOscillator Object
- 10.7 Audio File Location
- 10.8 DigiSynth Specifications
- 10.9 The CDigiSynthVoice Object
- 10.10 DigiSynth Files
- 10.11 DigiSynth: RackAFX
- 10.12 DigiSynth: VST3
- 10.13 DigiSynth: AU
- 10.14 Challenges

11 VectorSynth and AniSynth: Vector Synthesizers

- 11.1 The Vector Joystick
- 11.2 Vector Paths
- 11.3 Rotors and Orbits
- 11.4 Combining Paths and Orbits
- 11.6 The Moog AniMoog® Anisotropic Synthesizer
- 11.6 VectorSynth and AniSynth Path Modes
- 11.7 The Vector Joystick and JS Program in RackAFX

- 11.8 The Vector Joystick in VST3
- 11.9 The Vector Joystick in AU
- 11.10 Programming a Vector Path: VST3 and AU
- 11.11 Vector Synth Specifications
- 11.12 The CVectorSynthVoice Object
- 11.13 Vector Synth Files
- 11.14 VectorSynth: RackAFX
- 11.15 VectorSynth: VST3
- 11.16 VectorSynth: AU
- 11.17 AniSynth
- 11.18 AniSynth Specifications
- 11.19 The CAniSynthVoice Object
- 11.20 AniSynth Files
- 11.21 AniSynth: RackAFX, VST3 and AU
- 11.22 Challenges

12 DXSynth: FM Synthesizer

- 12.1 FM and PM Theory
- 12.2 FM Spectra
- 12.3 FM/PM Rules
- 12.4 Dynamic Spectra
- 12.5 DXSynth Specifications
- 12.6 FM and PM in Oscillator Objects
- 12.7 Yamaha DX100® vs. DXSynth
- 12.8 The CDXSynthVoice Object
- 12.9 DXSynth Files
- 12.10 DXSynth: RackAFX
- 12.11 DXSynth: VST3
- 12.12 DXSynth: AU
- 12.13 Challenges

13 Delay Effects

- 13.1 Circular Buffers
- 13.2 Delay Effect Topologies
- 13.3 The CDelayLine Object
- 13.4 The CStereoDelayFX Object
- 13.5 Using the CStereoDelayFX Object in your Plug-Ins
- 13.6 MiniSynth + FX: RackAFX
- 13.7 MiniSynth + FX: VST3
- 13.8 MiniSynth + FX: AU
- 13.9 Challenges

Appendix A

- A.1 Converting the VST3 Template Synth (9 Steps)
- A.2 Converting the AU Template Synth (10 Steps)