

VisualAudio—Accelerating the Development of Embedded Audio Products



VisualAudio—Accelerating the Development of Embedded Audio Products

Introduction

Engineers involved in the design and development of next generation audio products are being impacted by several market trends. First, the complexity of products is increasing due to the continued introduction of new audio formats, the need to play back several simultaneous content streams, and the steady migration of high end features, such as automatic room equalization, into consumer level products. Second, audio products are emerging that contain a significant amount of overall software complexity, such as network stacks. Examples include networked audio players and digital video recorders. Often software engineers without a solid background in audio processing or sound quality issues are charged with incorporating audio features.

This paper provides an overview of VisualAudio®, a graphical development environment from Analog Devices, Inc. (ADI) that is specifically aimed at audio development engineers. VisualAudio provides many of the software building blocks, together with an intuitive graphical interface, for designing, configuring, and tuning audio systems. VisualAudio consists of the following components:

Audio Module Library—A collection of optimized real-time audio processing functions sufficient for developing many different types of audio products. Separate versions exist for SHARC® and Blackfin® Processors.

VisualAudio Designer—A PC application for graphically creating audio processing networks. VisualAudio Designer can use audio processing functions supplied in the Audio Module Library or custom user-written functions.

Example Platforms—These real-time frameworks make it easy to develop systems on EZ-KIT Lite® evaluation hardware provided by Analog Devices. Source code is provided for all example platforms, and they serve as convenient starting points for product development.

VisualAudio works in conjunction with VisualDSP++®, the development and debugging environment for SHARC and Blackfin Processors. VisualAudio generates optimized product-ready code and simplifies developing complex audio processing software. VisualAudio benefits development engineers by reducing development cost, risk, and time to market. This paper discusses the features and benefits of VisualAudio, and describes how the tool is used during a typical product development cycle.

SHARC and Blackfin Processor Families

Digital signal processors (DSPs) have become the platform of choice for audio products because their programmability allows them to be quickly customized to a specific market or application. The SHARC and Blackfin Processor families from ADI are particularly well-suited for this task and are being used in consumer, automotive, and professional applications.

VisualAudio was originally written for the SHARC Processor, and its coverage has recently been expanded to include the Blackfin Processor. Currently, VisualAudio supports the SHARC ADSP-2126x and ADSP-2136x processor families as well as all Blackfin Processors. These two processors are quite different. The SHARC is a 32-bit, SIMD, floating-point DSP with a large internal memory. The Blackfin is a high-clock-rate, 16-bit, SIMD, fixed-point DSP with 32-bit embedded microcontroller capabilities (such as instruction and data caches, high performance external SDRAM interface, and significant general-purpose extensions to the instruction set). The Blackfin also has extensive power management features, making it suitable for portable battery-powered devices. Both processors come in a variety of models, many with integrated audio peripherals such as asynchronous sampling rate converters, S/PDIF transceivers, etc.

Despite their differences, both processors are supported by VisualAudio. Complementary audio module libraries allow processing networks to be easily moved between processors. In the SHARC implementation, the signal type is 32-bit floating-point; on the Blackfin, it is 32-bit (double precision) fixed-point. The SHARC is aimed at high end products requiring sophisticated audio processing, such as AVR receivers, digital automotive amplifiers, or professional studio equipment. The Blackfin is finding application in portable products such as MP3 decoders or networked audio players.

DSP Software Architecture

VisualAudio divides the DSP architecture into two separate portions. The *framework* is responsible for processor initialization, audio I/O, decoders, host communication, and control. The *layout* consists of the real-time audio processing functions, audio parameters and state variables, and a tuning interface for making parameter changes while audio is being processed. In VisualAudio, frameworks and layouts are, for the most part, decoupled, allowing layouts to be easily migrated between different frameworks.

On the DSP, the interface between the framework and the layout consists of buffers of audio data. These buffers are filled by the framework, processed by the layout, and then returned to the framework. Each buffer contains a block of audio data, and the processing is done in place by the layout. The processing is illustrated in Figure 1.

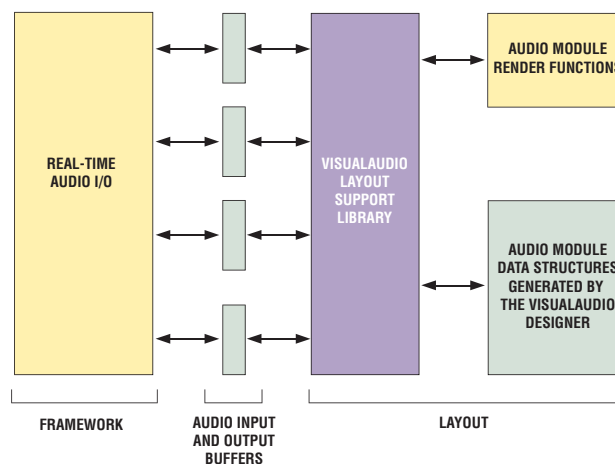


Figure 1. Overview of the real-time DSP architecture showing the interaction between the platform and the layout.

The division between framework and layout also exists on the PC, and *platforms* serve as interfaces between VisualAudio Designer, the framework, and VisualDSP++. This is illustrated in Figure 2. The platform encompasses a *Platform File* and a *Platform Project*. The Platform File is an XML file that describes the capabilities of the framework (and thus the target hardware) to VisualAudio Designer. For example, it lists the number of audio inputs and outputs, processor type, processor speed, etc. The Platform Project is a VisualDSP++ project that builds the final executable. It references the framework, source files generated by VisualAudio Designer, real-time audio processing functions, and related libraries. The Platform Project is managed independently outside of VisualAudio.

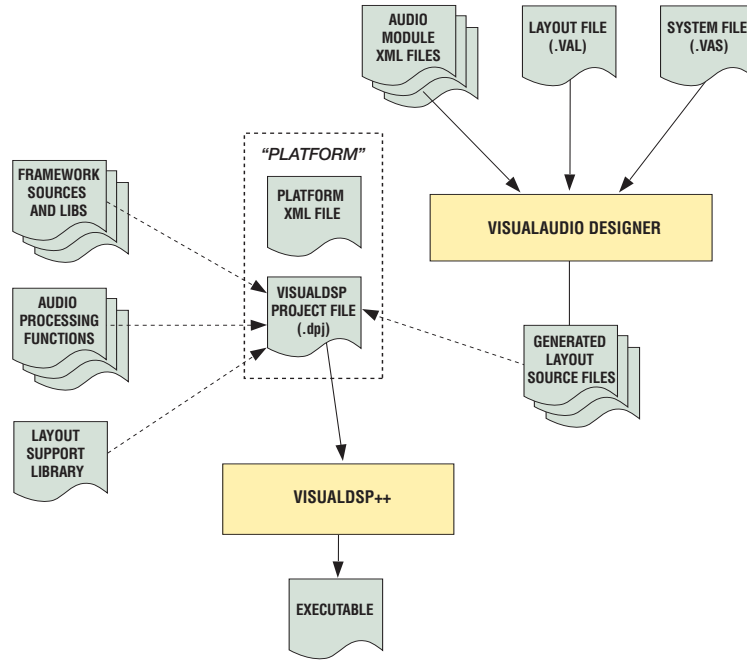


Figure 2. PC tool flow showing the relationship between VisualAudio Designer, VisualDSP++, and the platform.

Development Process

Different skills are needed to develop the platform and layout, and these tasks are often done by separate engineers. In the discussion that follows, we will refer to the person writing the framework code as the *embedded software engineer*, and the person developing the layout as the *audio engineer*. VisualAudio Designer is used to design, configure, and tune audio processing layouts, and will thus primarily be used by the audio engineer.

The steps involved in using VisualAudio for product development are shown in Figure 3 and described in detail below.

1. The audio engineer begins by selecting a hardware reference board that can be used to prototype and refine the audio processing. The reference board contains a superset of the memory and peripherals found in the final product, and can be used to start development of the audio layout prior to having actual target hardware. Analog Devices provides EZ-KIT Lites for most SHARC and Blackfin Processors. VisualAudio provides example platforms that operate on many of the EZ-KIT Lites. These platforms are actually complete systems that include I/O, buffering, an interface to the layout, and also a platform file.

2. The audio engineer designs the audio processing layout, including writing any custom audio modules. Inspectors are used to configure audio module parameters.
3. VisualAudio generates source files that describe the layout and, together with the example platform, run on the reference hardware.
4. In parallel, the embedded software engineer develops the framework for the target hardware. The framework initializes the processor, manages the I/O, and buffers audio into blocks. The embedded software engineer starts with one of the example platforms supplied with VisualAudio. Source code is provided, allowing them to be customized to the target hardware.
5. The embedded software engineer incorporates the VisualAudio Layout Support Library. The library provides the run-time interface to the audio processing and can be tested with a simple layout created with VisualAudio.
6. Once the framework is debugged, the embedded software engineer creates and publishes a platform file. This platform file enables VisualAudio Designer to generate layout code for the target platform. A platform project is also created, which ties together all of the DSP software components needed to build the executable.
7. At this point, the embedded software engineer and the audio engineer share files. This enables the audio layout designed by the audio engineer to execute on the target hardware.
8. The audio engineer can continue to use the full capabilities of VisualAudio on the target hardware. This includes updating the layout, tuning the system, and creating presets.

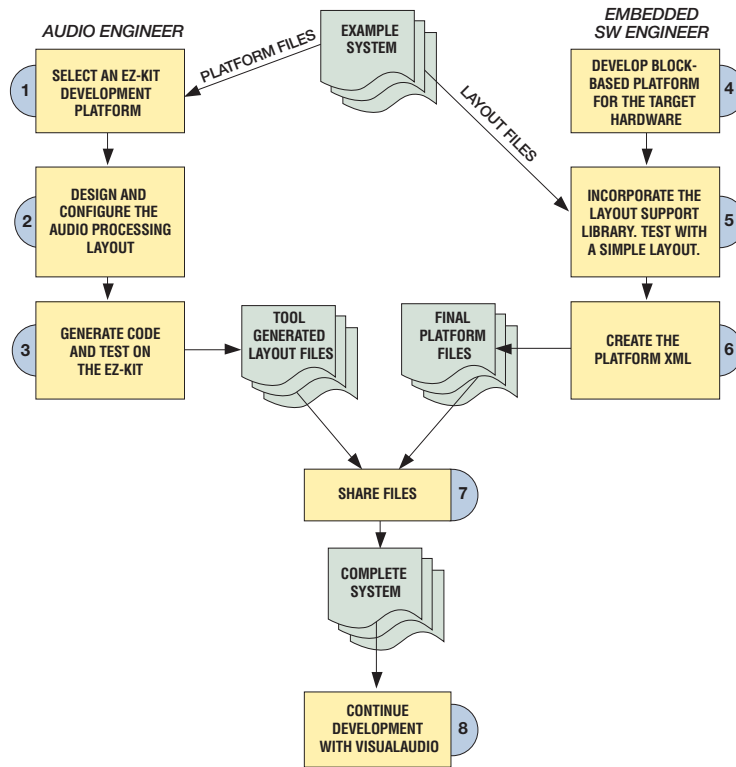


Figure 3. Typical development process used with VisualAudio.

Open and Extensible

Audio system developers have a significant investment in design techniques, tool chains, and audio algorithms. Introducing a new development tool can be time consuming and risky if it does not interface well with existing tools. VisualAudio was designed with this in mind, and offers several features that allow it to work in conjunction with other software tools and to be customized to a specific hardware platform.

The VisualAudio audio module library can also be extended through an open API. The user or third parties can create custom audio modules in C or assembly language, and these new modules are easily included within the VisualAudio design environment. Source code for the entire library of the ADI supplied audio modules is included together with detailed documentation and examples.

The VisualAudio software framework is based on a modular driver model. Well-defined interfaces for audio I/O, host communication, and audio decoders are provided. These functions can be customized by users for their particular hardware, and example source code is provided for ADI reference hardware.

Summary

The complexity of digital signal processing software within audio products continues to increase due to the introduction of new audio formats and the desire to offer unique and differentiating features. To address this, ADI has developed VisualAudio, an integrated audio software development environment. VisualAudio simplifies audio product development by providing engineers with many of the key software building blocks found in audio systems, together with an intuitive graphical interface. VisualAudio jump-starts product development; reduces development cost, time, and risk; and allows engineers to focus on innovation and product differentiation.



Analog Devices, Inc.
Worldwide Headquarters
Analog Devices, Inc.
One Technology Way
P.O. Box 9106
Norwood, MA 02062-9106
U.S.A.
Tel: 781.329.4700
(800.262.5643,
U.S.A. only)
Fax: 781.461.3113

Analog Devices, Inc.
Europe Headquarters
Analog Devices SA
17-19 rue Georges Besse
Antony, 92160
France
Tel: 33.1.46.74.45.00
Fax: 33.1.46.74.45.01

Analog Devices, Inc.
Japan Headquarters
Analog Devices, KK
New Pier Takeshiba
South Tower Building
1-16-1 Kaigan, Minato-ku,
Tokyo, 105-6891
Japan
Tel: 813.5402.8200
Fax: 813.5402.1064

Analog Devices, Inc.
**Southeast Asia
Headquarters**
Analog Devices
22/F One Corporate Avenue
222 Hu Bin Road
Shanghai, 200021
China
Tel: 86.21.5150.3000
Fax: 86.21.5150.3222

**Embedded Processing
and DSP Support**
U.S.A.:
processor.support@analog.com
Fax: 781.461.3010
Europe:
processor.europe@analog.com
Fax: 49.89.76903.157
www.analog.com/processors