

How a Graphical Audio Development Tool Can Simplify Embedded Audio Product Creation

By Paul Beckmann, Ph.D., DSP Concepts, and Gerard Andrews, Cadence Design Systems

Audio products are becoming more complex, as are consumers' expectations of them. Creating these products—from concept to R&D and tuning—can be difficult and time-consuming. In this paper, we will discuss the challenges in the product creation phases and explain how the combination of a graphical audio development tool with a low-power image processing DSP can transform what has long been an inefficient process.

Contents

Introduction	1
Dearth of DSP Engineers.....	1
Product Concept	3
Product Research.....	3
Product Development.....	3
Product Tuning.....	4
Techniques to Improve Audio Product Development	4
Graphical Development Tool for Faster Audio DSP Programming	4
Summary.....	5

Introduction

Audio has become an important function in a variety of applications. For example, a typical car today has a music and entertainment system (CD player, traditional and/or satellite radio, MP3 player, DVD player), a navigation system, BlueTooth connectivity with the smartphone, and typical alert-related sounds that chime when, for instance, the headlights are left on or the trunk door is ajar. In an electric car, the vehicle might synthesize engine noises to alert nearby pedestrians. In fact, a car now typically boasts 10 to 15 different audio algorithms controlling each of these different sound-related functions. Moreover, sounds are not simply released in a vehicle; voice command and control functions, such as hands-free calling, mean that always-on, always-listening capabilities are needed to act on user-generated sounds.

There are four main stages of embedded audio product development: concept, research, development, and tuning. Each of these stages must account for how each of the audio functions will work on their own and, in many cases, together. However, the overall development process hasn't changed much over the years—which means it hasn't improved. OEMs are generally so focused on getting the next product out the door that they've got little time to investigate tools that could streamline their process.

Let's take a closer look at the technical roles behind developing embedded audio products, as there are some knowledge/expertise gaps that are contributing to some of the challenges.

Dearth of DSP Engineers

The audio product development process involves a DSP engineer, an audio engineer, and an embedded software engineer. Now, there are plenty of embedded software engineers around, and it's not difficult to find talented audio engineers, either. However, it is hard to find qualified audio DSP

engineers—the ones who actually write the code to perform a specific function, such as noise cancellation or beamforming. See Figure 1 for a depiction of how rare it is to find an engineer with the skills of all three of these disciplines.

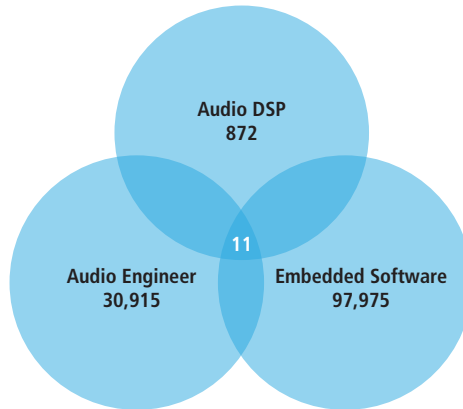


Figure 1: Based on a search of LinkedIn profiles, there's a relatively small number of audio engineers who have a background in DSP audio theory and embedded software design. Source: DSP Concepts

Audio DSP engineers must understand the math and algorithms behind each audio function. They need expertise with audio processing blocks such as EQ, echo cancellation, and noise suppression. Familiarity with the audio characteristics of speakers and microphones is required. It also doesn't hurt to have an understanding of acoustics design.

Compounding this problem is the sub-specialty of audio DSP, which takes into account not only the mathematical plotting of the different signal-to-noise ratio (SNR) and signal manipulations, but the understanding of the human psychoacoustic responses on how to handle those signal manipulations. A textbook understanding of DSPs alone is not good enough to deliver an optimal audio system.

Let's also note here that initial software implementations are typically written in generic C code. To achieve the desired performance, this code must be further optimized and, in older DSP architectures, even ported to assembly language. But there are fewer engineers studying the mathematical foundations of algorithms required to complete these optimizations. At the college level, students tend to be more interested in studying higher level languages, so graduates are entering the market with limited experience in low-level software.

This gap in knowledge/expertise can ultimately be detrimental to product quality. Often, DSP engineers end up overloaded and, as a consequence, are forced to make suboptimal design decisions. The audio engineers—who have the aural skills to evaluate audio quality—are often not involved in the process by the DSP engineers and, as a result, can't get exactly what they need for optimal sound. Designs are deemed to be "good enough" and get shipped in this condition (which explains why there are so many MP3 player docking stations on the market with subpar sound).

If you want to go beyond "good enough" quality with a development process that is smooth, reliable, and efficient, then you'll need tools and a methodology to repair what is now a broken process. Let's first take a closer look at each of the product development stages (Figure 2).

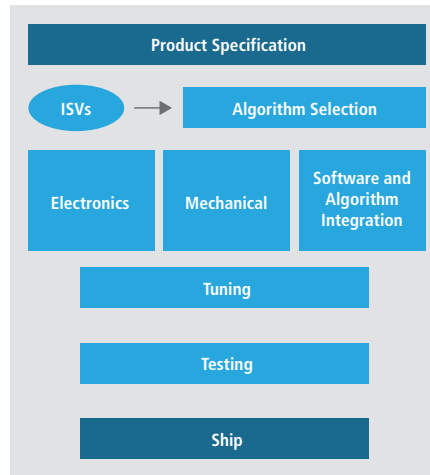


Figure 2: Traditional product development process

Product Concept

Product marketing and product engineering discuss and decide on top-level hardware- and software-based features in this earliest phase of product development. What will the new MP3 player need? Wireless headphones? Wireless speakers? Should it interface with a smartphone? What kinds of user controls should be integrated? Cost, time to market, and available resources to implement the product concept are key considerations at this stage.

Product Research

The DSP engineer gets more involved during the research phase. Here, the DSP engineer evaluates the hardware options and constraints in terms of criteria such as features, memory, cost, and power consumption. The engineer also determines what kind of software is needed to enable the desired functions, and whether the software exists or needs to be written. Third-party algorithms are an option here, as is in-house algorithm development. Time, effort, and cost are deciding factors.

Now, as we've noted, audio DSP engineers have a lot of responsibilities on their plates. Yet, performing a thorough IP evaluation is time consuming. For example, say you're considering noise-cancellation IP from a few third-party vendors. In addition to determining which IP will produce the best audio quality for your design, you'll need to answer, for each IP option, an array of questions that affect the overall design: What is the CPU load for the algorithms? What is the memory requirement? What clock speed is needed? This evaluation and decision process continues for each audio algorithm required to provide the desired functionality.

Typically, by the time this stage is underway, the processor for the design has already been chosen. Hardware and software development are happening in parallel, which impacts the end product. So, as you evaluate your IP options, you might find that the processor isn't fast enough, so you must give up features or sound quality. For instance, high-quality "upmixing," converting from a two-channel stereo file to a multi-channel surround output, requires careful manipulation of the audio signals without losing any information. To put that in perspective, by comparison, a dropped frame in a video application is usually unnoticeable to viewers. However, when it comes to audio, the ears are sensitive enough to notice the pops or clicks resulting from missing data. Similar decisions and tradeoffs must be made for each IP that you evaluate or create in house.

Product Development

At this phase, the embedded systems engineer has the chip integrated onto the board and conducts testing and validation exercises to ensure that everything works well together. This engineer's work continues as the board is integrated into the end product. Through this effort, the choices that the embedded engineer makes determine—and limit—which features will or won't make it into the final product. With traditional development methods, it's only after the process of fitting everything in that the engineer can determine if there is any processing power remaining to support any additional features.

Product Tuning

This phase is where the audio engineer—the one with the “golden ear”—ensures sound quality of the audio device by tweaking and tuning multiple parameters in real time. Different audio functions have different sample rates, and various components in the end product can impact sound quality. Consider our automotive example—you can’t really tune an automotive audio device until the car is fully manufactured. Even the presence of carpet can change the way that the sound sounds inside the vehicle.

Every product category requires some level of last-minute tuning. You need a way to guard against tuning and then breaking some aspect of the design.

Another consideration is that sound-quality comparisons are optimal when they are performed almost instantaneously. Often, however, DSP engineers must rebuild and compile code for different sound settings. So by the time an audio developer listens to a second or a third version, he or she likely won’t have the same fresh recollection of how the first one sounded.

Techniques to Improve Audio Product Development

As you can see, each phase of the product development process has challenges in terms of skill sets, time, and effort required. The skill sets required are unlikely to change any time soon. At the same time, development schedules are probably only going to get more aggressive, especially when designing for the ultra-competitive consumer market.

The right tools and methodologies can make all the difference in improving the embedded audio product development process. What is needed is a system that allows you to quickly design your audio system, prototype it on a board, test, and then deploy. Each engineer involved in the project must be able to collaborate, develop in parallel, and tune the design efficiently. They should have a fast way to evaluate and compare the performance of different audio algorithms on a target processor. Ideally, the design team should also be able to focus on design optimization and differentiation versus low-level coding.

Imagine how efficient this process could be if you had a graphical design environment that ties together the different engineering functions: audio engineering, DSP engineering, and embedded software engineering. In a graphical environment, individual functions can be represented by “blocks” of code that can be arranged and re-arranged by anyone on the team. Real-time tuning capabilities would be useful, so that code does not need to be changed and re-compiled before it can be heard again. Then, the audio engineer can optimize the sound of the DSP design. In addition, an ideal system would provide a large library of audio functions, pre-optimized for efficiency. This way, design teams would not be hampered by any gaps in expertise. An ideal solution should also allow the design team to quickly evaluate processing resources, audio algorithms, and IP in order to streamline the development cycle and, where possible, reduce development risks and costs.

Graphical Development Tool for Faster Audio DSP Programming

DSP Concepts offers a graphical development tool (Figure 3) for creating embedded audio software. The Audio Weaver® tool, recently optimized for the Cadence® Tensilica® HiFi DSPs for audio, video, and speech, can accelerate audio DSP programming by up to 10X, compared to traditional development methodologies. The tool provides this improved turnaround time because developers are able to work in parallel at different phases of the development cycle and to reuse pre-built, highly optimized audio processing algorithms. Tensilica HiFi programmers won’t have to write code or invoke a compiler in order to develop their audio applications. Instead, they can:

- Easily and quickly test different algorithms to select the best sound for their design
- Easily integrate software via tested and proven blocks of code
- Perform real-time tuning
- Perform regression testing, which mathematically ensures accuracy and pinpoints bugs in specific modules

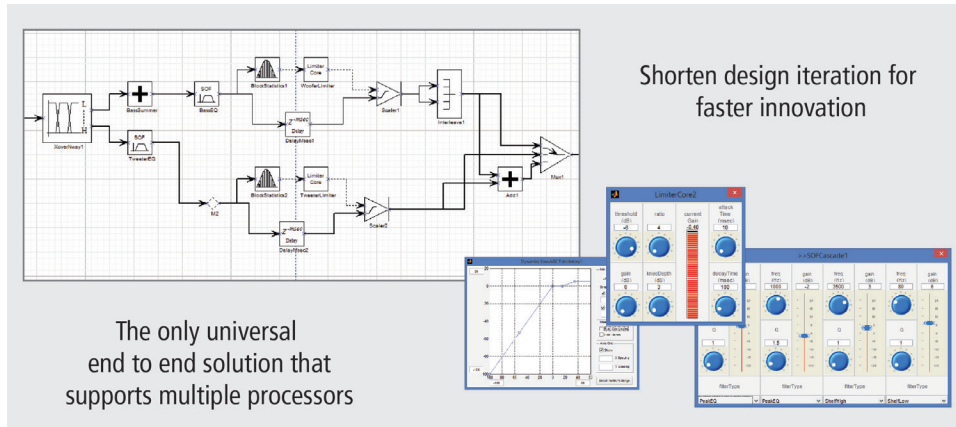


Figure 3: Audio Weaver from DSP Concepts is the market's only universal end-to-end embedded audio product creation solution for multiple processors.

The integration between Audio Weaver and Tensilica HiFi DSPs supports the market's growing need for wearable devices that feature an audio component ("hearables," if you will). For these types of applications, small form factor, low energy consumption, and always-on functionality are important—and these are areas where Tensilica HiFi DSPs have demonstrated strong results.

Summary

Audio signal processing requirements are as varied as the end applications providing the audio functionality, from always-on, always-listening smartwatches to automotive applications with multiple audio inputs. An embedded audio product creation methodology that allows parallel development, reuse of proven audio processing algorithms, and real-time tuning can help bring an outdated process up-to-date to meet 21st Century requirements and expectations.



Cadence Design Systems enables global electronic design innovation and plays an essential role in the creation of today's electronics. Customers use Cadence software, hardware, IP, and expertise to design and verify today's mobile, cloud and connectivity applications. www.cadence.com