

Ensuring Audio Signal Quality in Automotive Infotainment Systems

J. William Whikehart
Visteon Corporation

ABSTRACT

In automotive infotainment systems, multiple types of digital audio signals are usually present. Some come from internal sources, such as a CD or USB stick, and some come from external sources, such as an internet stream or digital radio. These sources usually have different sample-rates, and may also be different from one or more system sample-rates. Managing and transporting these signals throughout the system over different sample-rate domains require detailed upfront architecture analysis and correct system design to ensure signal quality is maintained to the desired level. Incorrect design can add significant user-perceivable noise and distortion. This paper examines the key analysis factors, the effects of poor design and the approaches for achieving robust signal handling and ensuring desired signal quality.

CITATION: Whikehart, J., "Ensuring Audio Signal Quality in Automotive Infotainment Systems," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(1):2013, doi:10.4271/2013-01-0163.

INTRODUCTION

Automotive infotainment systems are becoming increasingly complex, with a multitude of different audio signal paths, inputs, outputs and internally-generated signals. Signals must be moved and processed within the system to ensure desired signal quality levels are met. This becomes even more complicated in systems where signal data movement and processing are performed on general-purpose processors instead of dedicated DSP IC's. This paper presents the most important areas that the system designer must address to ensure robust audio signal routing and processing: sample-rate domains, sample-rate conversion, and audio signal stream buffering.

Although there are many papers and presentations that address the theory, uses and implementations of sample-rate conversion ([2][3][4] are just a few examples), there seems to be little information available that discusses the practical audio design issues found in complex infotainment systems, which will be addressed by this paper.

First to be presented is a typical automotive infotainment system, from an audio processing and routing perspective. Next, concepts on sample-rates are provided, followed by the main section which covers sample-rate domains and conversion as well as signal-stream buffering. A special sample-rate conversion issue that exists for certain types of audio sources is then discussed, followed by a summary.

AUDIO VIEW OF A TYPICAL INFOTAINMENT SYSTEM

Figure 1 presents a typical automotive infotainment system with its audio paths, inputs, outputs and internal sources. Audio sources usually include AM/FM radio, digital audio broadcasts (HD Radio®, satellite radio, European DAB and Digital Radio Mondiale™), CD, an auxiliary analog audio input jack, external digital audio sources (USB stick, personal music player, hands-free phone receive audio, hands-free phone audio streaming), and one or more microphone inputs for hands-free phone and adaptive noise reduction. Typical outputs are two or more speaker channels, possibly secondary headphone channels for rear-seat entertainment, and hands-free phone send audio.

External audio inputs and outputs may arrive and depart via vehicle busses like MOST and Ethernet. Internal sources may include a chime generator and playback of sound files for announcements or special sounds.

Each of the sources may have a different audio sample-rate. For example, the DAB, CD and some of the external digital audio sources may have rates of 22, 32, 44.1 or 48 ksp/s, while AM/FM audio may come in at a non-standard sample-rate, and hands-free phone receive audio may be at 8 or 16 ksp/s. Analog sources such as microphone inputs and auxiliary audio are likely converted to digital using the system sample-rate using A/D's. The speaker outputs usually come from an amplifier driven by D/A's running at the

system sample-rate. Hands-free phone transmit audio may use an 8 kbps or 16 kbps rate.

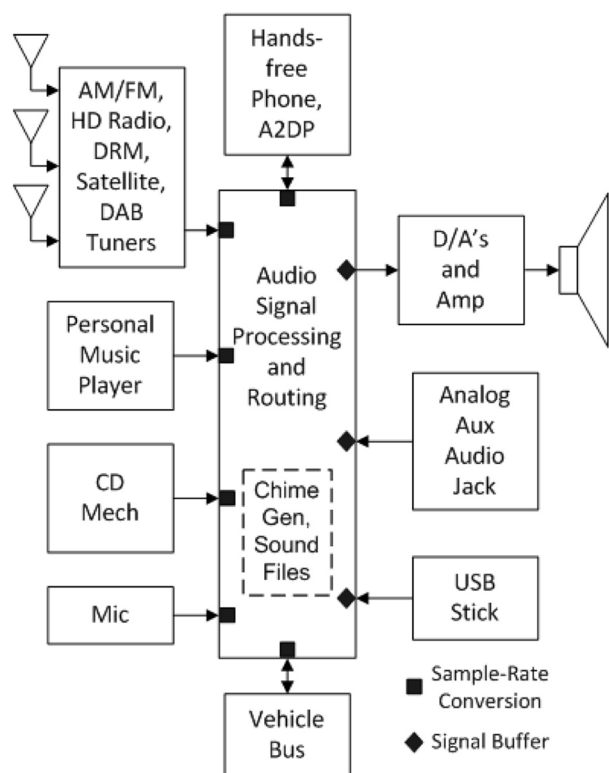


Figure 1. Audio View of a Typical Infotainment System

Within the system, many signal processing functions are performed, such as volume control, tone control, muting and unmuting, level control, mixing of multiple sources, and equalization. This processing operates at the system sample-rate, and this rate may be at a rate different from any of the source rates. It will certainly be different from those of some of the sources. Matching the system sample-rate to the particular source being played is usually not an option, because most of the many DSP coefficients used in the processing, such as those used in digital filters, are dependent on the sample-rate, and it is cumbersome to change large numbers of coefficients for a change in sample-rate. The choice of system sample-rate may also be constrained to available rates determined by the overall system clock processor clock and clock dividers, or and by other factors.

Figure 1 shows likely locations where audio signals move from one sample-rate domain to another, i.e. where sample-rate conversion must be performed. This is one of the most important elements of the signal paths. Also shown and just as important are signal buffers, which are used to move data between inputs, outputs, the signal processing and within parts of the signal processing.

It is common in many systems for samples to be grouped into packets of multiple consecutive samples, which are also called frames. For efficient use of processor cycles, packets rather than individual samples are moved and processed by the system. The length of a packet in time is equal to the

number of samples in the packet multiplied by the time for one sample, which is the inverse of the sample-rate for the particular signal being moved or processed.

The movement of signal data and the signal processing in an infotainment system may be performed on a general-purpose microprocessor rather than a dedicated digital signal processor (DSP) integrated circuit (IC). The microprocessor may be running a real-time operating system (RTOS) with multiple processes running at the same time. The particular points in time at which signal data is moved or processed will likely be variable and unpredictable to some degree. For example, packets from the audio decoder of a DAB audio source are produced according to when the audio decoder code task is actually executed. Because the RTOS determines when all tasks in the system are executed, and because it does this based on task priorities and on events happening throughout the system, the decoder code will be executed at irregular times and thus the decoder's output packets will appear at irregular times.

SAMPLE-RATES

Nominal and Locked Sample-Rates

Probably the most important aspect of infotainment system audio design is the location of time-bases, or master clocks, and corresponding master-slave sample-rate relationships. For any audio path location in the system and for audio signal sources generated outside the system, the audio samples exist at specific points in time determined by the particular master clocks that pertain to those locations and sources. A CD mechanism usually produces a stream of audio output samples at a rate that is derived from a clock oscillator in the mechanism. For a song coming from a digital audio broadcast, the values of the samples have been determined based on a sample clock derived from a master clock existing at the source of the broadcast. And the infotainment system's own system sample clock is derived from the system's clock oscillator. Obviously, these different master clocks cannot be synchronized, at least no without great difficulty.

Suppose the system is playing the audio from a personal music player using a digital connection. Audio samples are flowing through this connection. And suppose the audio player is playing from a compressed file that has a sample-rate after decompression of 44.1 kHz, so it is outputting samples at 44.1 kHz, probably accompanied by some kind of a sample clock signal. Furthermore, suppose the infotainment system sample-rate is 44.1 kHz, that is, it has been designed to process samples at 44.1 kHz based on its own master clock. Are the source and system sample-rates the same? NO! The sample-rate at the player output is derived (locked) to the master clock existing in the player. The sample-rate of the infotainment system playing the song on its speakers is derived (locked) to the system's master clock. These sample-rates can be said to be nominally the same, but it is impossible for them to be exactly the same, i.e. perfectly locked to each other, because they are each locked to

different master clocks. So, for the purposes of system design, they must be considered to be *different* sample-rates.

Sometimes a source includes metadata indicating the sample-rate of the audio material, or maybe the sample-rate of the source is known by definition, e.g. a regular CD is at 44.1 kHz. This is just the *nominal value* of the sample-rate at the source. This is not a physical clock that pinpoints in time when the samples from the CD exist - the nominal value is only a value that can be used by the infotainment system to know the approximate sample-rate at the source.

Two sample-rates that are locked together correspond to the frequencies of corresponding sample clocks that *must* have been derived from the same master clock via integer multipliers and dividers, or that must be synchronized through temporary locking means not practical for infotainment systems. This point of nominal rates vs. locked rates is very important because it shows that even for source and system sample-rates that are nominally the same, the audio signal will need to cross from one sample-rate domain to another when the infotainment system inputs the signal.

Master and Slave Sample-Rates and Clocks

The samples of a digital audio signal exist at certain points in time, and usually these are uniformly-spaced, where the time interval between points is equal to the inverse of the sample rate. For a signal source, a signal destination or a part of the system that controls the timing of its own samples, the source, destination or part of the system is usually labeled a sample clock master or sample-rate master for those samples. Conversely, when the timing of samples of a signal source, destination or part of a system is controlled by another part of the system, the source, destination or part of the system is usually labeled as a sample-rate or sample clock slave. This relationship between one part of a system that is a sample-rate master and another part that is a sample-rate slave is a relative master-slave relationship between that source and destination. Depending on the complexity of the system, the sample-rate master could also be a sample-rate slave to another part of the system. Using the CD example already described, the CD source might be the sample-rate master in relation to processing that performs some math on these samples, the processing being a sample-rate slave. But the processing might be the sample-rate master to a D/A that inputs the processed samples, the D/A being a sample-rate slave. A system could also have a source that is a sample-rate slave and a destination that is a sample-rate master. An example of this is a compressed music file on a USB stick, where the files on the stick are output from the stick as needed by the processing that performs decoding, and the decoder is producing samples at the system sample-rate. The USB stick is a sample-rate slave to the decoder, which is the sample-rate master in this relationship.

Note that a source of a signal stream need not be sample-rate master. As seen in the example above, it is also possible that a source is a sample-rate slave, where the sample stream

from the source is input by a destination that is a sample-rate master.

When a source is a sample-rate master, it is said to be “pushing” samples to the destination, a sample-rate slave. By pushing, the source is controlling the flow of samples. When a destination is a sample-rate master, it is said to be “pulling” samples from the source. By pulling, the destination is controlling the flow of samples.

CONNECTING IT ALL TOGETHER TO ENSURE AUDIO QUALITY

It is clear from the previous sections that the audio signals paths through infotainment systems can be complex. It is important that these paths move and process the audio samples in such a way that the quality of the signals represented by the samples is maintained to some desired degree. Audio quality is typically measured in terms of distortion and noise content. For example, people have been listening to AM/FM for many years while in vehicles. When the listener is tuned into his or her favorite station, the listener expects essentially perfect audio, at least to the typical limits of quality available for AM/FM. When the listener plays a CD, perfect high-quality audio is expected. And although there may be somewhat lower audio quality expectations for some digital audio sources, maybe due to compression being used, the listener still expects this quality level to be consistent.

Thus, certain levels of audio quality will be expected by listeners. In general, to meet these expectations, the audio samples must move along their paths continuously as uninterrupted streams. Isolated skipped samples or packets occurring even just occasionally may cause significant signal quality degradation. When a source is supplying a stream of samples faster than the stream can be consumed by its destination (an overflow condition), it may seem reasonable to drop samples or packets from the stream. When a destination is consuming a stream of samples faster than its source can supply them (an underflow condition), it may seem reasonable to repeat samples or packets, or even mute the audio for a short period of time to allow the source to catch up. These simple solutions can all cause significant audio quality degradation. In particular, dropping and repeating packets of audio, or inserting mutes that are the same lengths as packets, can cause severe distortion. The impacts of these simple solutions are discussed in more detail later in this paper.

Careful analysis and design of the system's audio paths is therefore critical. Given the audio quality requirements of the system, the key areas for the designer to consider are:

- Sample-rate domains and conversion
- Locations where asynchronous and synchronous sample-rate conversions are required
- Locations and lengths of buffers

Sample-Rate Domains and Conversion

Regardless of whether an audio sample stream is handled as individual samples or in packets, when an audio sample stream must pass from one sample-rate domain to another, the sample-rate of the signal must be converted using a sample-rate converter. The sample-rates of the two domains may either be relatively synchronous or asynchronous. If synchronous, they are derived from the same master clock and are locked together. If asynchronous, they are derived from different master clocks and are not locked together. In synchronous sample-rate conversion, the ratio of the input rate to the output rate can be expressed by the ratio of two integers. In asynchronous conversion, it cannot. Generally, asynchronous sample-rate conversion is more complex than synchronous sample-rate conversion.

A sample-rate converter inputs samples from a source at an input sample-rate and outputs samples to a destination at an output sample-rate. The source pushes its samples or packets to the sample-rate converter's input - it is in control at the input of the sample-rate converter, and so it is the sample-rate master of the input interface and the sample-rate converter is the sample-rate slave. The destination pulls the samples or packets from the sample-rate converter's output - it is in control of the output of the sample-rate converter, and so it is the sample-rate master of the output interface and the sample-rate converter is the sample-rate slave also at its output.

Let us review what sample-rate conversion really means. An audio signal is represented by a stream of audio samples, usually spaced uniformly in time. In order for the signal to be accurately represented, the sample-rate must be at least two times the bandwidth of the signal being represented. A signal can be represented by any set of samples and sample-rate meeting this rule, which is commonly referred to as the Nyquist sampling theorem [1]. So, sample-rate conversion means creating output samples at points in time corresponding to the output sample-rate such that the signal is faithfully represented by the output points, which is possible if the output rate meets the Nyquist sampling theorem. The output samples are calculated using the values of samples at the points in time corresponding to the input sample-rate and using knowledge of the relationship between the input and output sample-rates.

Sample-rate converters can take various forms, and they are usually not perfect - they will always reduce signal quality to some degree. Their designs can be simple to complex, depending on how much degradation is allowable.

A Simple Sample-Rate Converter

Probably the simplest approach to sample-rate conversion is to just repeat or delete samples in the stream as needed to keep the average input and output sample rates matched. [Figure 2](#) shows such a converter. It can be used for either synchronous or asynchronous sample-rate conversion. It consists of a serial FIFO buffer, input and output pointers,

and a controller. The buffer holds consecutive samples or packets. When a sample or packet arrives at the buffer input, being pushed to the buffer input by some source, it is written by the controller into the buffer at the location indicated by the input pointer, and the pointer is incremented either by one sample or by the length of the packet, modulo the length of the buffer. When a sample or packet needs to be output from the buffer, being pulled by the destination, the sample or packet is read by the controller from the buffer location indicated by the output pointer, and the output pointer is incremented either by one sample or by the length of the packet.

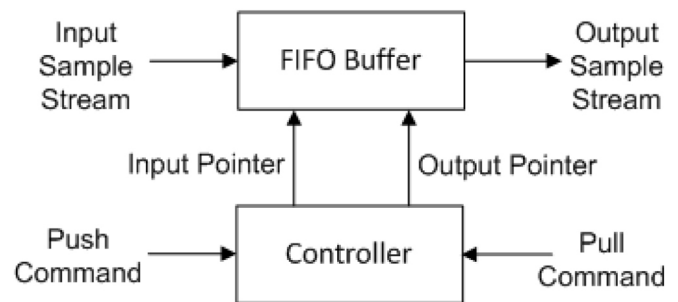


Figure 2. A Simple Sample-Rate Converter

Suppose the input sample-rate is faster than the output sample-rate. The incoming samples will overflow the buffer unless the number of samples is reduced somewhere at the buffer input, inside the buffer, or the buffer output. Conversely, if the input sample-rate is slower than the output sample-rate, the buffer will underflow, and will end up empty, unless the number of samples at the buffer input, inside the buffer, or at the buffer output is increased.

So, samples either need to be added or removed from the stream to prevent buffer underflow or overflow. Detection of underflow or overflow, or better yet detection of the buffer getting close to underflow or overflow, can be easily accomplished by monitoring the relative positions of the buffer input and output pointers, modulo the length of the buffer. If the input pointer is approaching the output pointer (overflow), a sample is removed from the stream. If the output pointer is approaching the input pointer, a sample is added to the stream. The simplest way to remove a sample is to delete one particular sample, and the simplest way to add a sample is to repeat an existing sample. This same approach can be used repeated samples.

Whether samples or packets are inserted or deleted, this process results in a signal stream with discontinuities at the insertion or deletion points. The insertion or deletion of a sample actually creates two types of discontinuity - one in amplitude and one in phase. Both of these contribute to the degradation of signal quality, adding distortion. The distortion may sound like audio pops, for lower insertion and deletion rates, and may sound like buzzing or noise for higher rates. The degradation increases with increasing insertions or deletions per second. If the converter inserts or deletes