



15-03-2023

Audio-DSP

Research document



Busse Lommers, Robin van den Dungen,
Mahmud Gürler, Silas Kamphuis, Hein
Verhallen, Youri Tils

BECREATIVE PROJECT AUDIO-DSP

Table of contents

Lists of tables, figures and abbreviations	3
1 Research questions	4
1.1 What is the best method for creating digital filters?.....	4
1.1.1 Different mathematical methods for digital filter design	4
1.1.2.....	6
1.2 What is the best method for creating digital effects?	6
1.3 What is the most suitable anti-aliasing filter?	7
1.3.1 Causes for aliasing.....	7
1.3.2 Anti-aliasing filters.....	8
1.4 What is the optimal needed roll-off for the anti-aliasing filter for a given bandwidth such that the noise can be negligible?.....	8
1.5 What is the minimum sample frequency needed to capture the desired frequency spectrum? ..	9
1.6 What is the minimum frequency range to be sampled to achieve sufficient detailed audio?..	9
1.7 What is the lowest allowable noise for decent audio?.....	9
1.7.1 Total Harmonic Distortion	9
1.7.2 Aliasing intermodulation distortion.....	10
1.7.3 SNR	10
1.7.4 Conclusion.....	12
1.8 What ADC resolution is needed such that the quantization error and noise level are on par?	14
1.9 What ADC and DAC architecture is most suitable for this application?	14
1.10 What kind of processor is most suitable for this application?	18
1.11 What is the permissible jitter for accurate audio?.....	18
1.12 What is the maximum allowable ripple on the reference voltage for the ADC and DAC?... 18	
1.12.1 ENOB (Effective Number of Bits).....	18
1.12.2 ADC Effective Resolutions	19
1.12.3 Delta-Sigma A/D Converters Resolution	19
1.12.4 Over Sampling Ratio (OSR).....	19
1.12.5 Usecase of the CS5532-BS (example).....	21
1.13 How much RAM does the system need?.....	22
1.14 How much flash does the system need?	23
1.15 What power supply topology is best suited for each part of the system?.....	23
1.16 How do we ensure that the USB input of the processor is recognized by Windows as an audio device?.....	26
2 References	27

Lists of tables, figures and abbreviations

List of tables

Geen gegevens voor lijst met afbeeldingen gevonden.

List of figures

Figure 1: Frequency spectrum of an ADC to visualize the required AAF magnitude response	9
Figure 2, Higher frequency gets mirrored around half the sample frequency and creates intermodulation distortion.....	10
Figure 3, Frequencies above half the sample frequency have full attenuation. Thus no aliasing intermodulation occurs.....	10
Figure 4, SNR examples.....	11
Figure 5, Frequency response examples.....	11
Figure 6 Dynamic range (SNR) and effective number of bits (ENOB), as functions of oversampling ratio (OSR) and modulator order (m).	20
Figure 7 Cirrus' CS5532-BS precision low-noise 24-bit converter.	20
Figure 8 Datasheet LDO ADR440	21
Figure 9 Datasheet CS5381 ADC.....	22
Figure 10: Global overview of the power supply section of an audio DSP.	24
Figure 11: Capacitance multiplier principle	26

Abbreviation	Explanation
DAC	Digital-to-analog converter
ADC	Analog-to-digital converter
SNR	Signal-to-noise ratio
THD	Total Harmonic Distortion
ENOB	Effective Number of Bits
SINAD	Signal-to-noise and distortion ratio

1 Research questions

The goal of this project is to research how to make an audio-DSP. This raises the main research question: **“How to design an audio-DSP?”**. From the main research question the following sub-research questions are derived:

1.1 What is the best method for creating digital filters?

1.1.1 Different mathematical methods for digital filter design

There are a lot of different mathematical methods for designing digital filters. Here is a large list of a lot of different methods generated by ChatGPT.

1. Fourier analysis: This method is used to decompose a signal into its frequency components. By analyzing the frequency components of a signal, it is possible to design digital filters that can remove unwanted frequency components.
2. Z-transform: This method is used to convert the difference equation of a digital filter into a transfer function in the Z-domain. This transfer function can then be used to analyze the frequency response of the filter and to design the filter.
3. Windowing: Windowing is a method used to design finite-duration impulse response (FIR) filters. This involves multiplying a rectangular window function with the impulse response of the desired filter to obtain a truncated impulse response.
4. Parks-McClellan algorithm: This is a method used to design optimal FIR filters. The algorithm seeks to minimize the difference between the desired frequency response and the actual frequency response of the filter.
5. Chebyshev approximation: This is a method used to design digital filters with a ripple in the passband or stopband. Chebyshev filters can be designed using the Chebyshev approximation method, which is based on Chebyshev polynomials.
6. Butterworth approximation: This is a method used to design digital filters with a maximally flat passband. Butterworth filters can be designed using the Butterworth approximation method, which is based on the Butterworth polynomial.
7. Sure, here are some more examples of mathematical methods that can be used in digital filter design:
8. Frequency sampling: This method is used to design finite impulse response (FIR) filters. It involves specifying the desired frequency response of the filter at a set of discrete frequencies, and then using the inverse discrete Fourier transform to obtain the filter coefficients.
9. State-space methods: State-space methods can be used to represent digital filters in a form that is amenable to analysis and design. The state-space representation consists of a set of first-order differential equations that describe the behavior of the filter over time.
10. Bilinear transform: The bilinear transform is a method for mapping analog filters to digital filters. This involves substituting the variable s with a function of the sampling frequency and applying the inverse Laplace transform to obtain the digital filter transfer function.
11. Impulse invariance: Impulse invariance is another method for mapping analog filters to digital filters. This involves sampling the impulse response of an analog filter and using it to obtain the impulse response of a corresponding digital filter.
12. Pole-zero placement: This method involves placing the poles and zeros of a digital filter in specific locations to achieve a desired frequency response. This method can be used for both infinite impulse response (IIR) and FIR filters.
13. Optimization methods: Optimization methods can be used to design digital filters that meet certain design specifications. These methods involve formulating a cost function that measures the deviation between the desired and actual frequency response of the filter, and then using optimization algorithms to minimize the cost function.

14. Frequency transformation: Frequency transformation is a method used to transform the frequency response of a digital filter while preserving the filter's stability. This method involves applying a mathematical function to the frequency variable of the filter transfer function.
15. Prony's method: Prony's method is used to estimate the parameters of an IIR filter from a finite set of samples. The method involves fitting an exponential signal model to the sampled data to estimate the filter coefficients.
16. Wavelet analysis: Wavelet analysis is a method used to analyze signals in the time-frequency domain. This method can be used to design digital filters that are well-suited for signals with non-stationary frequency content.
17. Singular value decomposition: Singular value decomposition is a method used to analyze the frequency response of a digital filter. This method involves decomposing the filter transfer function into a set of eigenvalues and eigenvectors.
18. Genetic algorithms: Genetic algorithms are optimization methods that are inspired by biological evolution. These algorithms can be used to design digital filters that meet certain design specifications.
19. Kalman filtering: Kalman filtering is a method used to estimate the state of a dynamic system from a set of noisy measurements. This method can be used to design digital filters that are well-suited for systems with non-stationary noise.
20. Multirate signal processing: Multirate signal processing involves the manipulation of signals at different sample rates. This method can be used to design digital filters that are optimized for specific processing tasks, such as decimation, interpolation, and filter banks.
21. Complex variable methods: Complex variable methods can be used to analyze the behavior of digital filters in the complex frequency domain. This method involves extending the frequency variable to the complex plane and using complex analysis techniques to analyze the filter transfer function.
22. Lattice structures: Lattice structures are a class of digital filter structures that can be used to implement both IIR and FIR filters. These structures can be designed using mathematical algorithms that optimize the filter coefficients.
23. Nonlinear optimization methods: Nonlinear optimization methods can be used to design digital filters that meet specific design constraints, such as passband ripple and stopband attenuation. These methods involve formulating the filter design problem as a nonlinear optimization problem.
24. Interpolation methods: Interpolation methods can be used to design digital filters that interpolate between sampled data points. These methods involve constructing a continuous function that passes through the sampled data points and then using the function to design a digital filter.
25. Matrix factorization methods: Matrix factorization methods can be used to analyze the frequency response of digital filters. These methods involve decomposing the filter transfer function into a set of matrix factors that can be used to analyze the filter's stability and performance.
26. Least squares: Least squares is a method used to estimate the coefficients of a digital filter from a set of input and output data. This method involves minimizing the sum of the squared errors between the predicted and actual output data.
27. Recursive least squares: Recursive least squares is a method used to estimate the coefficients of a digital filter from a stream of input and output data. This method updates the filter coefficients in real-time as new data becomes available.
28. Maximum likelihood: Maximum likelihood is a method used to estimate the parameters of a digital filter from a set of input and output data. This method involves finding the parameters that maximize the likelihood of observing the input and output data.
29. Bayesian inference: Bayesian inference is a method used to estimate the parameters of a digital filter from a set of input and output data. This method involves computing the posterior probability distribution of the filter parameters given the input and output data.

30. Wiener filtering: Wiener filtering is a method used to design digital filters that minimize the mean squared error between the desired output signal and the filtered input signal. This method involves computing the optimal filter coefficients using the input and output statistics.
31. Convex optimization: Convex optimization is a method used to design digital filters that meet specific design constraints, such as passband ripple and stopband attenuation. This method involves formulating the filter design problem as a convex optimization problem.

1.1.2

There are several methods that we can use to design a digital filter, the choice depends on the factor such as the filter type, complexity, and **performance requirements** in digital audio, sound waveforms are represented by samples. An analogue-to-digital converter (ADC) measures, or samples, analogue signal and assigns a digital value to each sample. Humans can typically hear frequencies between 20 Hz and 20 kHz. (A Hz is a cycle per second.) To adequately represent this frequency range digitally, the ADC needs to sample the audio waveform at least twice the highest audible frequency; hence we have the common sample rates of 44.1 kHz and 48 kHz.

- Finite Impulse Response (FIR) filters: These filters are easy to implement and provide linear phase response, but they can be computationally expensive for long filter lengths.
- Infinite Impulse Response (IIR) filters: These filters are computationally efficient and can achieve sharper roll-off characteristics, but they can introduce phase distortion and instability if not designed properly.
- Frequency sampling method: This method involves sampling the frequency response of an ideal filter and then using inverse FFT to obtain the impulse response.

References:

<https://eclipseaudio.com/fir-filter-guide>

1.2 What is the best method for creating digital effects?

- Adding **effects** to a DSP involves implementing digital signal processing algorithms that **modify the audio signal in some way**. There are many different types of effects that can be implemented, ranging from simple equalization and filtering to more complex effects like reverb, delay, distortion, and modulation.
 - To implement these effects, we would need to design digital signal processing algorithms that operate on the audio signal in real-time. This might involve using digital filter design techniques to create frequency responses that shape the audio in a desired way or using time-domain techniques like convolution to create effects like reverb or delay.

Design methods are:

- Delay-based effects
 - Delay-based effects involve creating one or more delayed copies of the input signal and then modifying the delay time or feedback amount to create the desired effect. Examples of delay-based effects include reverb, echo, and chorus.
- Filter-based effects
 - Filter-based effects involve using digital filters to modify the frequency content of the input signal. Examples of filter-based effects include EQ and phaser.
- Modulation-based effects

- Modulation-based effects involve using various modulation techniques, such as amplitude modulation or frequency modulation, to alter the characteristics of the input signal.
Examples of modulation-based effects include tremolo, vibrato, and ring modulation.
- Distortion-based effects
 - Distortion-based effects involve intentionally distorting the input signal to create a specific sound or tone.
Examples of distortion-based effects include overdrive, fuzz, and distortion.
- Pitch-based effects
 - Pitch-based effects involve altering the pitch or tuning of the input signal.
Examples of pitch-based effects include pitch shifting, harmonization, and auto-tune.

1.3 What is the most suitable anti-aliasing filter?

1.3.1 Causes for aliasing

<https://ptolemy.berkeley.edu/projects/chess/eecs124/reading/LeeAndVaraiya11.pdf>

<https://www.soundonsound.com/sound-advice/q-what-aliasing-and-what-causes-it>

An alias occurs when a signal above half the sample rate is allowed into, or created within, a digital system. It's the anti-aliasing filter's job to limit the frequency range of the analogue signal prior to A-D conversion, so that the maximum frequency does not exceed half the sampling rate — the so-called Nyquist limit.

Aliasing can occur either because the anti-alias filter in the A-D converter (or in a sample-rate converter) isn't very good, or because the system has been overloaded. The latter case is the most common source of aliasing, because overloads result in the generation of high-frequency harmonics within the digital system itself (and after the anti-aliasing filter).

The sampling process is a form of amplitude modulation in which the input signal frequencies are added to and subtracted from the sample-rate frequency. In radio terms, the sum products are called the upper sideband and the subtracted products are called the lower sideband. In digital circles they are just referred to as the 'images'.

These images play no part in the digital audio process — they are essentially just a side-effect of sampling — but they must be kept well above the wanted audio frequencies so that they can be removed easily without affecting the wanted audio signal. This is where all the trouble starts. The upper image isn't really a problem, but if the lower one is allowed too low, it will overlap the wanted audio band and create 'aliases' that cannot be removed.

Let's consider what occurs if we put a 10kHz sine-wave tone into a 48kHz sampled digital system. The sampling process will generate additional signal frequencies at 58kHz ($48 + 10$) and 38kHz ($48 - 10$). Both of these images are clearly far above half the sample rate (24kHz), so can be easily removed with a low-pass filter, which is the reconstruction filter on the output of the D-A converter, leaving the wanted audio (the 10kHz tone) perfectly intact. See Figure 1, above.

However, consider what happens if our 10kHz tone is cranked up too loud and overloads the A-D converter's quantising stage. If you clip a sine wave, you end up with something approximating a square wave, and the resulting distortion means that a chain of odd harmonics will be generated above the fundamental. So our original 10kHz sine wave has now acquired an unwanted series of strong harmonics at 30kHz, 50kHz and so on.

Note that these harmonics were generated in the overloaded quantizer and after the input anti-aliasing filter that was put there to stop anything above half the sample rate getting into the system. By overloading the converter, we have generated 'illegal' high-frequency signals inside the system itself

and, clearly, overloading the quantizer breaks the Nyquist rule of not allowing anything over half the sample rate into the system.

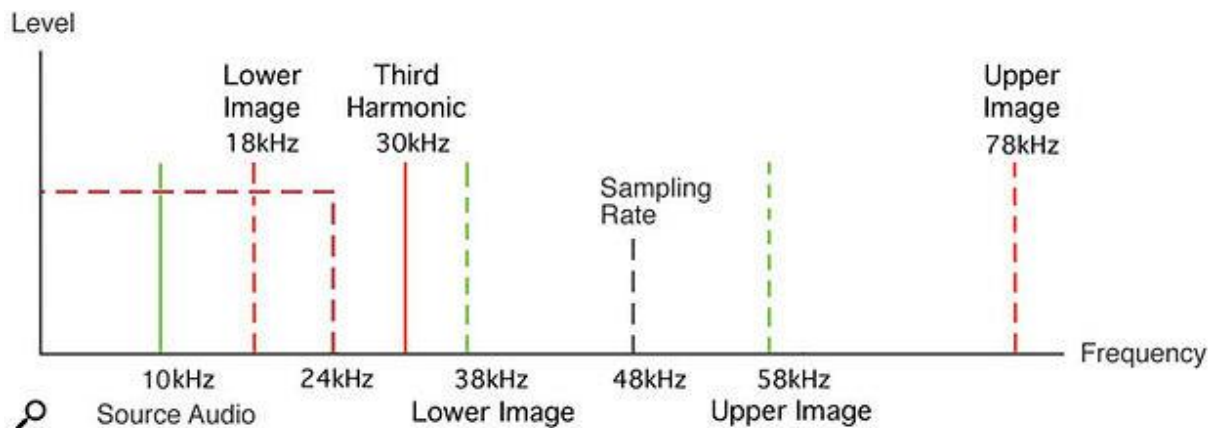


Figure 1, Aliasing example

Considering just the third harmonic at 30kHz for the moment, the sampling modulation process means that this will 'mirror' around the sample rate just as before, generating additional signal frequencies at 78kHz (48 + 30) and 18kHz (48 - 30). The 18kHz product is clearly below half the sample rate, and so will be allowed through by the reconstruction filter. This is the 'alias'. We started with a 10kHz signal, and have ended up with both 10kHz and 18kHz (see Figure 2, above). Similarly, the 50kHz harmonic will produce a 2kHz frequency, resulting in another alias.

Note that, unlike an analogue system, in which the distortion products caused by overloads always follow a normal harmonic series, in a digital system aliasing results in the harmonic series being 'folded back' on itself to produce audible signals that are no longer harmonically related to the source.

In the simplistic example I've explained, we have ended up with aliases at 2kHz and 18kHz that have no obvious musical relationship to the 10kHz source. This is why overloading a digital system sounds so nasty in comparison to overloading an analogue system.

1.3.2 Anti-aliasing filters

ANTI-ALIASING FILTER THEORY

A/D Converters are usually operated with a constant sampling frequency when digitizing analog signals. By using a sampling frequency (F_s), typically called the Nyquist rate, all input signals with frequencies below $f_s/2$ are reliably digitized.

1.4 What is the optimal needed roll-off for the anti-aliasing filter for a given bandwidth such that the noise can be negligible?

The needed roll-off of the AAF (anti-aliasing filter) depends on the bandwidth of the system, the maximum allowable noise floor of the system and the sample rate.

The maximum roll-off bandwidth can be determined by subtracting the chosen sampling frequency of the system by the two times BW of the system. As can be seen in the formula below:

$$BW_{(roll-off)} = f_s - 2BW_{system}$$

The above formula shows that for a given BW, at a low sample rate, an AAF is needed with a very steep roll-off. If a higher sample rate is used for the same BW, an AAF with a lower roll-off is permissible.

The lower the required roll-off of the AAF, the lower the order of the AAF can be. A lower order AAF will always introduce less distortion than a higher order AAF. As far as the AAF is concerned, the highest possible sample rate is desired for a given BW.

Using this maximum roll-off bandwidth together with the maximum allowable noise floor of the system, the minimum required roll-off of the AAF can be determined using the following formula:

$$\text{roll of } f_{min}[\text{dB/dec}] = \frac{SNR}{\log_{10}\left(\frac{f_s - f_{max}}{f_{max}}\right)} \text{ with } SNR \text{ in dB}$$

$$\text{roll of } f_{min}[\text{dB/oct}] = \frac{SNR}{\log_2\left(\frac{f_s - f_{max}}{f_{max}}\right)} \text{ with } SNR \text{ in dB}$$

Equation 1, Anti-aliasing filter minimal roll-off needed to achieve a given SNR

Figure 2 shows a graphical representation of the AAF in the frequency spectrum of an ADC.

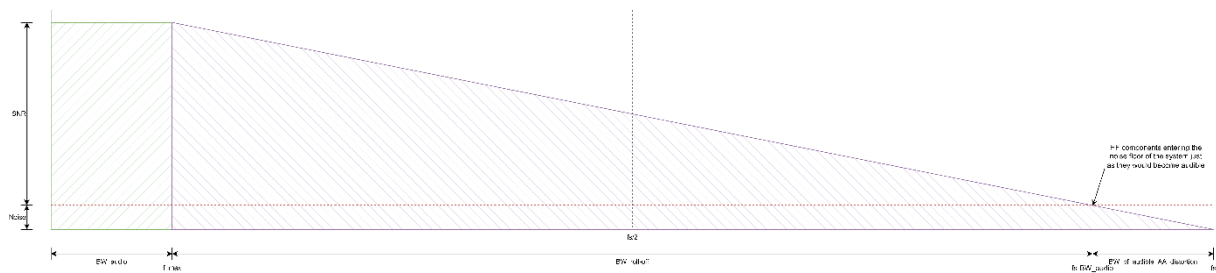


Figure 2: Frequency spectrum of an ADC to visualize the required AAF magnitude response

So according to Equation 1 by increasing the sample frequency we decrease the minimal roll-off needed for a given SNR. Therefore it would be best to choose the highest possible sample frequency.

1.5 What is the minimum sample frequency needed to capture the desired frequency spectrum?

The human hearing frequency range is 20Hz to 20kHz. According to the Nyquist theorem the sample frequency must be twice the highest measurable frequency to be able to reconstruct the original signal $f_s > 2 \cdot f$. This would mean that the sample frequency should be at least $f_s = 20\text{kHz} \cdot 2 = 40\text{kHz}$.

1.6 What is the minimum frequency range to be sampled to achieve sufficient detailed audio?

The best frequency response for high-quality audio devices is 20Hz to 20kHz [1].

1.7 What is the lowest allowable noise for decent audio?

1.7.1 Total Harmonic Distortion

Total Harmonic Distortion (THD) is the ratio between the sum of the power of all harmonics in a signal to the power of the fundamental frequency of the signal (see Equation 2 and Equation 3) [2].

$$THD(\text{dB}) = 20 \log \frac{\sqrt{H_2^2 + H_3^2 + H_4^2 + \dots + H_n^2}}{H_1}$$

Equation 2, Total Harmonic Distortion in dB

$$THD(\%) = \frac{H_2^2 + H_3^2 + H_4^2 + \dots + H_n^2}{H_1^2} \cdot 100\%$$

Equation 3, Total Harmonic Distortion in percent

1.7.2 Aliasing intermodulation distortion

When signals are sampled at frequencies higher than half the sample frequency, the signal gets mirrored around half the sample frequency [3]. This causes the mirrored signal to intermodulate with the fundamental signals and create unwanted harmonics (see Figure 3).

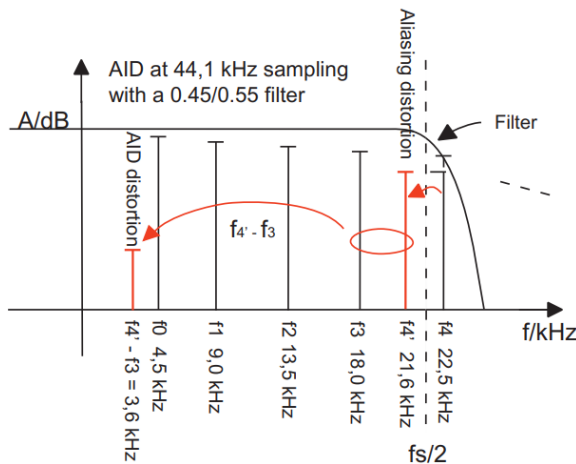


Figure 3, Higher frequency gets mirrored around half the sample frequency and creates intermodulation distortion.

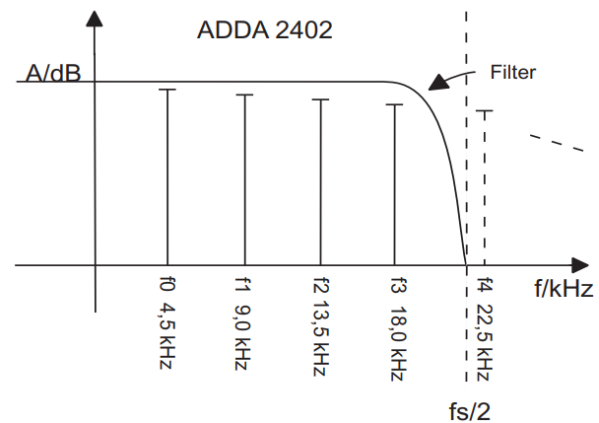


Figure 4, Frequencies above half the sample frequency have full attenuation. Thus no aliasing intermodulation occurs.

The solution to prevent aliasing modulation distortion is to have the aliasing filter have full attenuation at half the sample frequency (see Figure 4).

1.7.3 SNR

Any electronic audio device (amplifier, preamp, CD player, turntable, etc.) generates more or less interference signals and background noise which are affecting the sound signal [4]. It is the very functioning of its components (capacitors, transistors, resistors, etc.) that generates this noise.

The signal-to-noise ratio (SNR, see Figure 5) is expressed in decibels (dB) and indicates the level difference measured between the background noise generated by the circuits / components of a device (hiss when no signal passes through it) and the signal at its nominal level (defined at +4 dBu on analog devices, which corresponds to the “0” level of the VU meters of amplifiers, for example).

The higher the signal-to-noise ratio of a device, the lower its background noise, in other words the more “silent” its operation. Quieter operation makes more subtle sounds more audible.

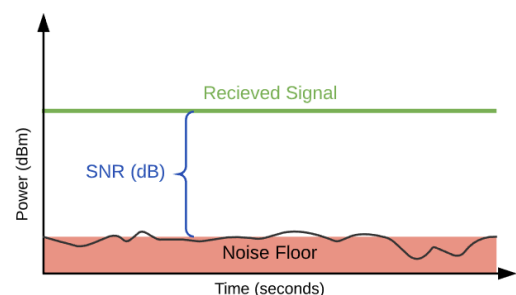


Figure 5, Signal-to-Noise representation

It is generally considered that a good signal to noise ratio is 60 dB or more for a phono turntable, 90 dB or more for an amplifier or CD player, 100 dB or more for a preamp.

Audio quality measurement primer paper: <https://www.eeweb.com/wp-content/uploads/articles-app-notes-files-audio-quality-measuerment-primer-1302041910.pdf>

AUDIO APPLICATION	SNR (DB)
Plain Old Telephone System or POTS	40
AM Radio, LP Records	50
FM Radio, Cassettes	70
CD Player	90
Human Ear Range (Instantaneous)	85
HiFi Studio Recording Equipment	120
Human Ear range (Total)	120

Figure 6, SNR examples

So according to this paper the total Human hearing range is 120dB. Therefore it would be best if the SNR is higher than 120dB [5].

Abbreviations: FS = Full Scale

Idle Channel Noise

Idle channel noise, in dB, is a ratio of the full scale RMS signal level to the RMS sum of all other spectral components over the specified bandwidth using a zero amplitude signal. ICN is a measure of the static noise produced by the device with a zero amplitude signal. It is useful to have good ICN performance in order to avoid an irritating noise commonly referred to as hiss, random noise, or white noise. Leakage from local sources, such as power lines, will show up and degrade ICN performance as well.

Frequency Response

Frequency response is the variation of the system's output level relative to the input level across a frequency range (EQ. 3). The typical human hearing range is 20Hz to 20kHz. Therefore, audio systems with a flat response over this range seem to sound the most natural. As a performance benchmark, an in-band frequency flatness tolerance, in dB, is usually specified. For example, most conventional stereo audio systems specify a frequency flatness tolerance of ± 3 dB. In addition, the -3dB points of the system at high and low frequency extremes are also specified.

AUDIO APPLICATION	FREQUENCY RESPONSE
Plain Old Telephone System or POTS	300Hz to 3kHz
AM Radio	100Hz to 5kHz
FM Radio	50Hz to 15kHz
Consumer Stereo System	20Hz to 20kHz
Professional Audio Equipment	5Hz to 24kHz

Figure 7, Frequency response examples

The -3dB point corresponds to the frequency at which the signal power has been reduced to one half of the normal 0dB signal level. A typical system frequency response plot is pictured in figure 1 with the -

3dB points at 20Hz and 20kHz. It has been shown that human perception is capable of detecting signal amplitude differences as low as 0.1dB. However, under dynamic signal conditions, 1dB is more realistic. Table 2 lists some common frequency response examples. The frequency response of a system is usually measured with an input signal that is -20dB below full scale.

$$\text{Frequency response } (f) = 20 \log \frac{V_o(f)}{V_i(f)} \text{ (EQ. 3)}$$

Decibels

Why Decibels? The decibel is a logarithmic unit which is used in a number of scientific discipline to compare some quantity with some reference value. Usually the reference value is the largest or smallest likely value of the quantity. We use decibels for several reasons. Quantities of interest often exhibit such huge ranges of variation that a dB scale is more convenient than a linear scale. For example, the listening level of your home stereo can vary by seven orders of magnitude. In addition, the human ear interprets loudness on a scale much closer to a logarithmic scale than a linear scale. The ‘bel’, named after Alexander Graham Bell, is the logarithmic ratio of two powers, P1 and P2, and is expressed in Equation 4.

$$A(B) = \log \frac{P_1}{P_2} \text{ (EQ. 4)}$$

The reference power is usually based on some very small natural occurrence. An example of which is sound pressure level. The reference sound pressure level in air is 0.0002 Pa(RMS). A typical loudspeaker, with 1 watt of power at 1kHz, produces 9B of sound pressure measured at a standard distance of 1 meter. The same loudspeaker is most likely limited to about 11B of pressure. Average listening levels are usually around 6B. This makes the measurement range only 5B. Due to the small size of the bel, the more convenient decibel has become standard and is expressed in Equation 5.

$$A(dB) = 10 \log \frac{P_1}{P_2} \text{ (EQ. 5)}$$

The decibel is often used to express the relative amplitude or power of a signal. Since power is the square of amplitude, a simple translation from Amplitude to Power can be made as in Equation 6. Some common variations on the decibel are listed below.

$$10 \log \frac{A_1^2}{A_2^2} = 20 \log \frac{A_1}{A_2} \text{ (EQ. 6)}$$

“dB FS” stands for “dB relative to full scale” and is generally used in all audio measurements unless otherwise specified. “dBr” stands for “dB relative to some absolute level.” When the absolute level is set to full scale, “dBr” is equivalent to “dB FS.” “dBV” stands for “dB relative to 1V.”

1.7.4 Conclusion

So for generic amplifiers such as CD player a typical SNR is 90dB. The total human ear range is 120dB. Therefore the SNR of HiFi audio system is also 120dB. Thus for a decent audio system a SNR of 90dB or higher is already sufficient. But to deliver a very good quality High-end product the SNR should be 120dB.

But with higher bit resolution of the ADC certain signal-to-noise ratios can be neglectable. This is because with a certain amount of ADC resolution the steps are bigger than the actual noise. For example at 24 bit and 1V reference you have a resolution of approximately 60nV. This is very small, therefore allowing a lot of noise to be measured as well. Thus to attenuate the noise enough for this resolution you would need a SNR of at least: $SNR = 20 \log \frac{1V}{60nV} \approx 144dB$. The typical SNR for common ADC resolutions are also shown in paragraph 1.8.

As the SNR needs to be extremely high, which is extremely difficult to achieve, you probably just go with an SNR of 120dB. This causes a lot of noise to be measured with the 24 bit resolution. Which actually means that you can't use all the 24 bits due to the noise. The bits that can be used are described as Effective Number Of Bits (ENOB) [6]. The equation for this term is seen in Equation 4.

$$ENOB = \frac{SINAD - 1.76}{6.02}$$

Equation 4, ENOB with the Signal-to-Noise And Distortion ratio

$$ENOB = \frac{SNR - 1.76}{6.02}$$

Equation 5, ENOB with the Signal-to-Noise ratio

SINAD stands for Signal-to-Noise And Distortion ratio. It is the ratio between the power of the whole signal and the noise and the distortion of the signal (see Equation 6).

$$SINAD = \frac{P_{signal} + P_{noise} + P_{distortion}}{P_{noise} + P_{distortion}}$$

Equation 6, SINAD

Note that Equation 4 assumes a full-scale input signal. Therefore when the signal level is not full-scale the ENOB calculation needs correction.

$$ENOB = \frac{SINAD_{Measured} - 1.76 + 20 \log \frac{Full\ scale\ amplitude}{Input\ amplitude}}{6.02}$$

Equation 7, ENOB without full-scale signal

So for example, with a SNR of 100dB we get an $ENOB = \frac{100dB - 1.76}{6.02} = 16.32\ bits$. Thus with this attenuation of 100dB it would be useless to have 24 bits as we would never use more than 16.32 bits. A resolution of 16 bits would be the best choice as you then use every bit.

With a SNR of 120dB we get an $ENOB = \frac{120dB - 1.76}{6.02} = 19.64\ bits$. This means that with a resolution of 24 bits we can't use 4.36 bits.

So to effectively use a 24 bit resolution we would at least need a SNR of 120dB to make a difference in resolution compared to a 16 bit resolution.

Signal to Noise and Distortion 'from Measurement'
(This can be replaced by SNR depending on situation)

Effective Number of Bits

$$ENOB = \frac{SINAD - 1.76}{6.02}$$

$10 \log_{10}^{(3/2)}$

$20 \log_{10}^2$

Quantization Error per 1 bit

1.8 What ADC resolution is needed such that the quantization error and noise level are on par?

Maximum quantization error = $\frac{V_H - V_L}{2 \cdot 2^n}$, where n is the number of bits.

Quantization error is the difference between the analog signal and the closest available digital value at each sampling instant from the A/D converter [7]. Quantization error also introduces noise, called quantization noise, to the sample signal. The higher the resolution of the A/D converter, the lower the quantization error and the smaller the quantization noise. The relationship between resolution (in bits) and quantization noise for an ideal A/D converter can be expressed as $Signal\ to\ Noise = -20 \log \frac{1}{2^n}$ where n is the resolution of the A/D converter in bits. S/N is the signal to noise and is expressed in dB. This relationship can also be approximated as $Signal\ to\ Noise = 6 \cdot n$. Typical S/N ratios for ideal A/D converters are 144dB for 24 bits, 96dB for 16 bits, 72dB for 12 bits, and 48dB for 8 bits.

1.9 What ADC and DAC architecture is most suitable for this application?

Article: <https://www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converters-the-successive-approximation-reg/#:~:text=The%20SAR%20ADC%20does%20the,state%20of%20the%20current%20bit.>

ADC

An Analog to Digital Converter (ADC) is a system that can convert analog signals into digital signals. In the case of this project: analog audio signals.

Continuous-time and amplitude analog signals are converted to discrete-time and amplitude digital signals. During this project data is lost, and noise gets added. The signal-to-noise ration (SNR) is determined by the accuracy of the ADC. This accuracy depends on the matching of the quantization levels and the true analog signal. Another factor that characterizes an ADC is the bandwidth. If the bandwidth of the ADC is twice the bandwidth of the analog signal, perfect representation of the signal is possible.

To make the signal not cut off on low levers an ADC can implement dither. This will add a small amount of noise so that there will be more information stored than before.

The sampling rate of the ADC is also important. The sample rate is the rate of which a continuous time signal is sampled. As mentioned before, if the sample rate is more than twice the amount of the highest frequency, perfect reproduction is possible.

Types of ADCs

Direct conversion (flash) – A direct conversion ADC uses multiple comparators that all have a certain voltage range that the comparator works in. Each comparator compares the input voltage to a reference voltage and converts it to a digital signal using a bit encoder.

This ADC architecture had poor energy dissipation and uses a lot of comparators to accomplish a wide range of signal.

Successive approximation – A Successive approximation ADC uses a binary search algorithm that finds the input voltage and compares it with an internal DAC (see info DAC) to converse the analog signal to a digital signal. This type of ADC is slower because this architecture of conversion tests all the weights of the algorithm and only one conversion is processed during n comparison cycles.

Ramp compare – This ADC generates a sawtooth signal that ramps up (or down) and quickly goes back to the original state. While doing this, the ADC compares the value of the ramping signal to the input signal. During this process a timer runs and stops when the signal of the input matches the ramp signal. The number of pulses the counter gave is the digital representation of the analog signal.

Wilkinson – The Wilkinson ADC charges a capacitor till it reaches the input voltage. The time it takes to discharge the capacitor is the digital representation of the amplitude of the input signal.

Integrating – With an Integrating ADC, the input voltage is ran through a integrator and can run up for a certain time. A known reference voltage than is ran through the integrator with reversed polarity. The input voltage is then functioned with the reference voltage, run-up time and run-down time. The time of the run-up and run-down is the digital output.

With this ADC, the resolution gets better the longer the ADC can run up. So, the ADC gets slower if the application is more demanding.

Delta encoded – The input voltage and an up-down counter DAC go through a comparator that controls the counter from the DAC. The counter changes until the DAC signal and input signal are the same. The counter is in this case the digital output.

Delta converters have a wide range and high resolution. Although, it depends on how the input signal behaves.

Pipelined – A pipelined ADC has several stages. A sample and hold stage, an ADC (e.g., a flash ADC) and a DAC stage. The input signal is sampled and hold for a certain amount of time. It is then converted to a digital signal where the most significant bits are then converted back by the DAC. This process can repeat several times if necessary. For example, a 4-stage, 16-bit ADC only needs 60 comparators.

These ADCs are very fast, have a high resolution and can be implemented efficiently.

Delta-sigma – This architecture uses a feedback loop together with an analog filter. This ADC architecture has potential for high resolution, high accuracy, and low noise. This architecture works by oversampling the analog input signal and using the feedback loop to make a high-speed digital bitstream. Delta-sigma ADCs are used in high-end audio equipment.

Time-interleaved – A time-interleaved ADC exists of multiple parallel ADCs that sample every x cycles. This ADC architecture exists to correct time interleaving mismatch errors.

Intermediate FM – An ADC with an intermediate FM stage uses a voltage-to-frequency converter to convert the analog input signal into an oscillating frequency. Then a frequency counter is used to read this frequency and output a digital signal.

Time-stretch – This ADC can digitize a wide bandwidth analog signal that can normally not be converted by a conventional ADC. A time-stretch ADC slows down the input signal and compresses the bandwidth. A normal ADC can now capture this slowed-down signal.

The ADCs that are used the most in audio applications are: Sigma-delta, Successive approximation and pipelined.

DAC

A Digital-to-Analog converter converts digital signal into analog signal. Architectures of different DACs is also determined by factors like resolution and sampling frequency. DAC's can make a signal worse so choosing the right architecture is important for this project. So in essential, a DAC does the reverse thing an ADC does.

An audio DAC is a low frequency, high resolution type DAC. Higher resolutions means more levels of an analog signal can be reached, thus a better audio quality.

Types of DACs

There are a couple of types of DACs frequently used in the audio world. Each has different focusses. Some DACs have the focus on the least significant bit, and others have the focus on the most significant bit. Sometimes, more than one type of architecture is used to help fix certain properties. Segmented DACs are DACs that are designed out of different architectures for specific performances.

The binary weighted – The binary weighted DAC has several bit inputs that lead into an operational amplifier where all the inputs have a powers-of-two weighting with the current voltage. This conversion method is very fast but has poor accuracy. Another disadvantage is that the power dissipation of a binary weighed DAC is very high.

The R2-R ladder (or binary ladder) – The R2-R ladder DAC only uses two values of resistors (which makes accuracy better and fabrication easier) and is easily scalable. Because of this, de DAC does work slower than the binary weighted DAC. With this type of DAC there is some delay due to switching based on the inputs.

Delta-Sigma – In the ADC section there was a sigma-delta ADC. The delta-sigma DAC is the exact opposite. It uses an interpolation filter to increase the sample rate and to reduce the time, so that the sample frequency is increased by four times. Then the signal goes through a modulator which acts as a

High-Pass Filter for the quantization noise and as a Low-Pass Filter to the signal. After this process the high-speed bit-stream is converted to analog by a 1-bit DAC. The signal is converted serially.

This DAC is one the fastest and most precise DAC.

Pulse-Width Modulation (PWM) – The PWM DAC uses a fixed frequency signal to create a series of pulses with the width proportional to the amplitude of the digital signal. This is a relatively simple DAC that is very accurate.

There are also oversampling DACs. These DACs take the original signal and oversamples it to a higher frequency. After this step the signal is filtered to remove noise. These DACs can achieve a higher precision and quality signal.

Segmented DAC

As said before, to fulfill certain specifications, multiple DACs are joined together as a segmented DAC. For example: A thermometer-coding DAC is used for the most significant bit, and a binary-weighted DAC is applied for the least significant bit.

Best DAC and ADC architectures for audio use

For this project, ADC- and DAC architectures should be chosen for audio signals. The project's aim is to have high quality audio processed so the ADCs and DACs should be capable of converting high quality signals into converted high quality signals. The minimum required sample rate is 44.1 kHz. But higher sample rates such as 96 and 192 kHz are favored. The ADCs and DACs of choice should be at least 16 bits, but 24 bits is preferred. The SNR depends on how many bits the ADC and/or DAC has. For a 24 bits component the SNR should be close to 144dB and for 16 bits close to 96dB.

There are a couple of brands that make audio ADCs and DACs. Texas Instruments and Asahi Kasei Microdevices are two big sellers of ADC and DAC chips. Because of the data available, and the widely available chips that they offer, Texas Instruments is the preferred company for this project.

Texas Instruments offers a wide spectrum of ADCs and DACs with the price ranging from a few euros to several hundred euros.

For this project with these specific requirements (and depending on stock) there can be made a list of good choices.

ADC

TI PCM18xx Series (24 bit, low power, 96-192kHz)

This ADC series has a lot of different models but are all ranging from 96 to 192kHz. The PCM18xx's are Delta-Sigma converters and are designed for audio processing application such as amplifiers, receivers, and audio recorders. The 1800 series is inexpensive (around 3 to 8 euros.)

DAC

TI PCM17xx Series (24 bit, 96-192kHz)

The TI PCM17xx series is a 24 bits DAC and is comparable with the 18xx series from the ADC. It's fast, precise, and inexpensive. These DAC chips are also built on the Delta-Sigma architecture. Cost of a chip is around 3 to 10 euros.

The 16 bits DACs and ADC cost the same as the 24 bits but are noticeably slower and have a lower SNR.

1.10 What kind of processor is most suitable for this application?

For DSP applications most of the time an FPGA or dedicated DSP is used. They both are great and will work for our application.

An DSP is mostly used in market segments that have a very specific use case. The specifications for these applications are very clear and will stay the same for the whole development process. This is because the hardware architecture is inflexible. This makes them unsuitable for dynamic applications. General-purpose DSPs cost between 1- and 300-euro dependent on their features and applications. In general, an DSP is easy to implement, and use. Most DSPs use C code and are well documented.

FPGAs are designed with logic elements and memory that are configurable. Therefore, an FPGA can be used for a wide range of applications of which digital signal processing. This also means that an FPGA is more modular so the end user can implement multiple hardware designs to ensure the FPGA fits in to specific requirements. FPGAs are equipped with embedded memory, embedded processors such as ARM Cortex and DSP blocks. The DSP blocks are there to perform specific DSP tasks that the normal logic elements cannot handle efficiently. In general FPGAs are more expensive around 100 to 1000 euro. There are less expensive FPGAs but there are generally not used for DSP applications.

1.11 What is the permissible jitter for accurate audio?

Jitter explanation: <https://www.stereophile.com/reference/1093jitter/index.html>

1.12 What is the maximum allowable ripple on the reference voltage for the ADC and DAC?

In order to understand the fundamental aspects of working with ADC's:

[Tips on Improving ADC Measurement Accuracy and Resolution Part 1](#)

[Understanding Analog-to-Digital Converter \(ADC\) Specifications - Developer Help \(microchipdeveloper.com\)](#)

1.12.1 ENOB (Effective Number of Bits)

[8] Analog-to-Digital Converter (ADC) resolution can be used to describe the general performance of an ADC. Resolution and accuracy are terms that are often interchanged.

The resolution of an A/D converter (ADC) is specified in bits and determines how many distinct output codes (2^n) the converter can produce. In other words, a resolution is the smallest voltage increment corresponding to a 1 LSB change. It's an important ADC specification because it determines the smallest analog input signal an ADC can resolve.

For example, an 8-bit ADC produces 2^8 or 256 output codes. The accuracy of the ADC determines how close the actual digital output is to the theoretically expected digital output for any given analog input. In other words, the accuracy of the converter determines how many bits in the digital output code represent useful information about the input signal. The accuracy of the ADC is a function of its internal circuitry and noise from external sources connected to the ADC input. In some cases, extra bits of resolution that are beyond the accuracy of the ADC can be beneficial.

1.12.2 ADC Effective Resolutions

[8]Effective resolution describes the useful bits from an Analog to Digital conversion with respect to the input noise. Effective Number of Bits (ENOB) is often used to specify ADC effective resolution. ENOB is not the same as the actual resolution of the ADC. Effective resolution is expressed using two units of measure, the specification of bits rms (0.707) refers to the output data. Effective resolution predicts the probability of a conversion level of repeatability of 70.7% for an input signal.

1.12.2.1 Calculation

ENOB equation (based on an ideal ADC's Signal-to-Noise Ratio (SNR): $SNR = (6.02) (ENOB) + 1.76 \text{ dB}$ where N is the ADC's resolution. $ENOB = (SNR - 1.76)/6.02$.

$$LSB = \frac{V_{cc} - V_{ref}}{2^n}$$

$$LSB = \frac{3}{2^{24}} = 178nV$$

$$Passband-Ripple = \pm 0.05dB$$

$$Passband-Ripple \text{ range} =$$

$$(3 * 0.05dB) = 3.01731918901$$

$$(3 * -0.05dB) = 2.98278022186$$

$$SNR = 99dB$$

$$ENOB = \frac{SNR - 1.76}{6.02}$$

$$ENOB = 16 \text{ bits}$$

$$Resolution = 16bit$$

$$Steps = 24bits$$

1.12.3 Delta-Sigma A/D Converters Resolution

[8]Delta-Sigma ADCs can provide resolutions as high as 24 bits. A given 24-bit Delta-Sigma converter may only provide 16 bits of accuracy. In this case, the 8 LSB's represent random noise produced in the converter. However, these noise bits are used with digital filter algorithms to increase the useful measurement resolution at the expense of a lower sampling bandwidth.

Effective Number of Bits (ENOB) of a Delta-sigma ADC is a function of the ADC's resolution in bits and its standard deviation:

$ENOB = N - \log_2(\sigma)$ where σ = standard deviation of data and N = number of ADC bits.

1.12.4 Over Sampling Ratio (OSR)

[9]Both from plausibility arguments in the time domain, and from explicit calculation in the frequency domain, it appears that the modulator circuit holds the key to the sigma-delta ADC's performance; that is, its ability to quantize an

analog input signal with a resolution considerably greater than the oversampling ratio. Furthermore, that figure of merit ($N_{eff}/\log_2 OSR$, where N_{eff} is the effective number of bits in the quantized digital output) grows with modulator complexity: contemporary ADCs employ "higher-order" modulators, meaning that the single difference amplifier and integrator is replaced with several cascaded stages of difference amplifier plus integrator, each driven from the common bitstream (see Figure 13.55).⁸¹ Higher-order modulators are widely used, because they extend the dynamic range without having to

increase the oversampling ratio (see below); they also suppress to a large extent the idle tones (see §§13.9.9 and 13.9.10) that afflict first-order modulators. Although our time-domain musings above may be helpful (if only to make plausible the excellent dynamic ranges that are claimed), any serious analysis must use

the frequency-domain approach. The latter shows that a higher-order modulator (constructed with m integrators) modifies the noise shaping such that the in-band quantization noise (i.e., dc to f_{\max}) is suppressed as $\text{OSR}^{m+0.5}$, where m is the modulator order ($m=1$ for Figure 13.51). Put another way, each doubling of the oversampling ratio suppresses quantization noise so as to increase the dynamic range by $m + 12$ bits; or, to state it in terms of modulator order, the effective number of bits (ENOB) is the \log_2 of the oversampling ratio (e.g., 6 for $\text{OSR}=64$) multiplied by $m+12$ (thus, for example, $\text{ENOB} \approx 15$ for a second-order modulator with oversampling ratio of 64). The graph in Figure 13.56 shows the theoretical maximum dynamic range of a delta-sigma ADC as a function of oversampling ratio and modulator order.⁸²

Another technique for extending dynamic range, speed or both, is to design a modulator that generates a modulated “wordstream” that is more than one bit wide. In Figure 13.51, for example, the 1-bit ADC, 1-bit DAC, and 1-bit register would be replaced with analogous 2-bit (4-level) components. There are lots of clever tricks to address imperfections in the multibit converters within the modulator (for example, cyclically exchanging bit positions to average out nonlinearities caused by offsets); they are well beyond the scope of this book.

Delta-sigma converters are adored by the professional audio community, owing to their combination of resolution,

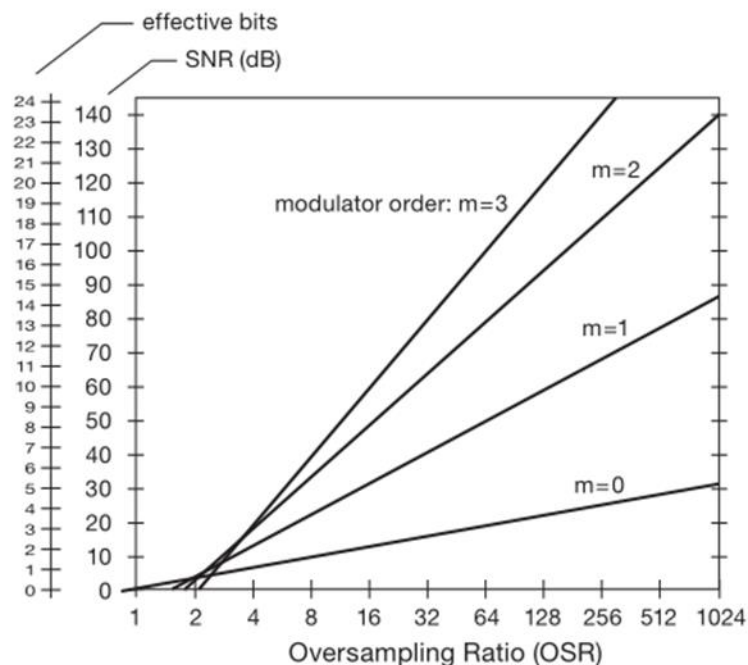


Figure 8 Dynamic range (SNR) and effective number of bits (ENOB), as functions of oversampling ratio (OSR) and modulator order (m).

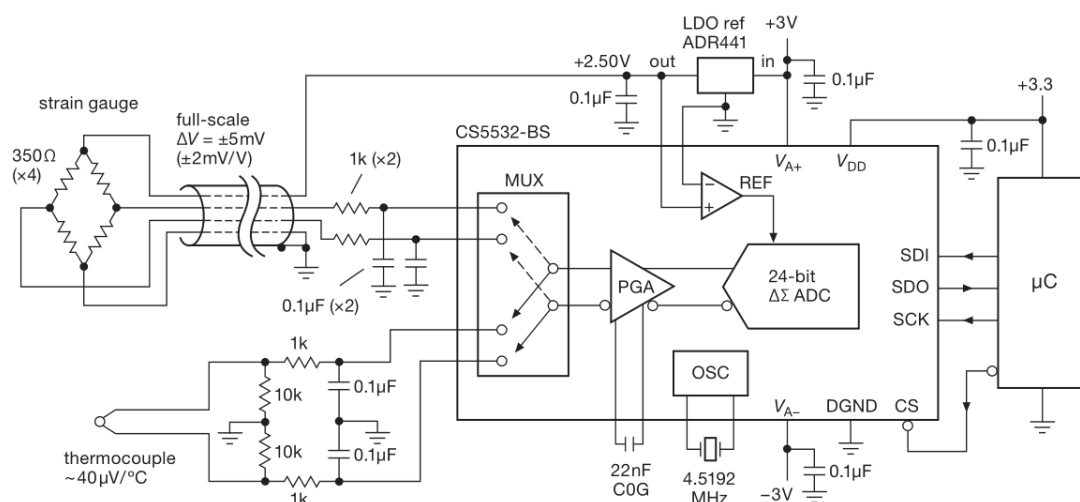


Figure 9 Cirrus’ CS5532-BS precision low-noise 24-bit converter.

inherent anti-aliasing, noise shaping, and monotonicity. (See Table 13.10.) If you open up pretty much anyone's high-quality audio gear, you'll probably find a circuit based around a 24-bit, 192 ksps 128× oversampling delta–sigma ADC; and chances are it will be a part from Cirrus (e.g., the CS5381) or AKM (e.g., the AK5394A). These parts seem to have “long legs” – they have been around for many years and represent good value in terms of price and performance. Audio ADCs universally use delta–sigma technology, but they differ greatly from their industrial ADC counterparts in the delta–sigma table. They generally have poor gain accuracy (5% to 10%) and dc offset (~25 mV), in part because these don't matter in the audio field. On the other hand, they do offer 0.1 dB or 1% stereo-channel gain matching. They're usually wired with ac-coupling, and they also have an internal digital highpass filter (typically ~1 Hz).⁹⁶ They are meant to operate at specific audio data rates, and they employ specialized audio PCM data output interfaces (I2S, TDM, etc.). Compared with industrial ADCs they have high latencies (data-output delays) of 12 to 63 sample intervals, even though they may advertise “low latency” (by which they mean small compared with, say, one millisecond of time delay). They have unique audio specifications, like A-weighted SNR, and spectral analysis-derived THD+N distortion specs. Figure 13.68 Shows a straightforward signal conditioning circuit, adapted from AKM's evaluation board, not unlike the innards of many commercial audio digitizers. The 5534 op-amp seems to be the perennial favorite (it's been around for at least three decades), inexpensive and “good enough.” Although you can do better in terms of distortion, what seems to matter most to audiophiles is dynamic range (set by the ADC resolution and the noise level); harmonic distortion at 0.001% is inaudible. However, we prefer full differential signal-conditioning circuits for high-performance audio ADCs, as described expansively in Chapter 5 (see Figures 5.70 and 5.102).⁹⁷ See §5.14.2E for more about pro-audio signal levels.

1.12.5 Usecase of the CS5532-BS (example)

FEATURES

Ultralow noise (0.1 Hz to 10 Hz)

ADR440: 1 μV p-p

ADR441: 1.2 μV p-p

ADR443: 1.4 μV p-p

ADR444: 1.8 μV p-p

ADR445: 2.25 μV p-p

Superb temperature coefficient

A grade: 10 ppm/°C

B grade: 3 ppm/°C

Low dropout operation (supply voltage headroom): 500 mV

Input range: ($V_{\text{OUT}} + 500 \text{ mV}$) to 18 V

High output source and sink current

+10 mA and –5 mA, respectively

Wide temperature range: –40°C to +125°C

Figure 10 Datasheet LDO ADR440

DC ELECTRICAL CHARACTERISTICS

(GND = 0 V, all voltages with respect to ground. MCLK=12.288 MHz; Master Mode)

Parameter	Symbol	Min	Typ	Max	Unit
Power Supply Current (Normal Operation)	VA = 5 V IA	-	36	43	mA
	VL, VD = 5 V ID	-	36	46	mA
	VL, VD = 3.3 V ID	-	24	28	mA
Power Supply Current (Power-Down Mode) (Note 8)	VA = 5 V IA	-	100	-	uA
	VL, VD = 5 V ID	-	100	-	uA
Power Consumption (Normal Operation)	VA, VL, VD = 5 V	-	360	445	mW
	VA = 5 V; VL, VD = 3.3 V	-	260	307	mW
	(Power-Down Mode)	-	1	-	mW
Power Supply Rejection Ratio (1 kHz) (Note 9)	PSRR	-	65	-	dB
VQ Nominal Voltage		-	2.5	-	V
Output Impedance		-	25	-	kΩ
Maximum allowable DC current source/sink		-	0.01	-	mA
Filt+ Nominal Voltage		-	5	-	V
Output Impedance		-	4.5	-	kΩ
Maximum allowable DC current source/sink		-	0.01	-	mA

Figure 11 Datasheet CS5381 ADC

LDO output is 1μV p-p, this means even before the ADC the reference voltage will have a very low ripple on the output. The CS5381 ADC itself has a typical PSRR of 65dB.

1.12.5.1 Calculation

As calculated before the LSB for a 24-bit ADC with a 3V reference voltage is 178nV (see header 1.12.2.1 Calculation). Therefore, the reference voltage must be below the LSB voltage. The combination of the LDO ADR440 and the CS5381 ADC this is possible.

$$ADC \text{ input after PSRR} = \frac{1\mu V_{P-P}}{65dB} = \frac{1\mu V_{P-P}}{10^{-3.25}} \approx 563pV$$

1.13 How much RAM does the system need?

The product should be able to store some data. This is needed for some of the audio effect we want to make. The size of the data is dependant of the sample rate, amount of bits and time u want to safe. The following formula can be used to calculate the size:

$$\text{Filesize} = \frac{\text{sampling frequency} * \text{bit depth} * \text{length}}{8}$$

The answer of this formula will be the file size in bytes. If we want to make an high quality audio DSP we probably need to use a sampling frequency of 192KHz and a bit depth of 24bit. The time is dependant on the amount of delay we want to add. Lets say for now we want to be able to store 6 seconds of audio data.

$$\text{Filesize} = \frac{192.000 * 24 * 6}{8} = 3456000 = 3.456 \text{ MB}$$

This is the amount needed to store 6 seconds of mono audio.

If we want to have 6 channels of audio each able to store 6 sec of data the total needed RAM will be.

$$3.456 \text{ MB} * 6 = 20.736 \text{ MB}$$

1.14 How much flash does the system need?

Depends on the program size.

1.15 What power supply topology is best suited for each part of the system?

The most suitable power supply topology depends on the application. For example, switch mode power supplies have the advantage that they can achieve a high efficiency and therefore generally dissipate less power than linear voltage regulators. I say generally because for applications requiring very little power, linear voltage regulators can still be more efficient because they have lower idle losses than switch mode power supplies.

Furthermore, switch mode power supplies produce noise on the power supply lines. This noise can be significantly reduced by proper design of switch mode power supplies, but switching power supplies are still not the most suitable power supplies to use for powering noise sensitive components.

It is best to use a linear voltage regulator to power noise-sensitive components since these linear power supplies don't generate switching noise. The audio DSP will have to use a combination of switch mode power supplies and linear voltage regulators. This is to be able to use the benefits of both.

For example, a switch mode power supply will be necessary to create a negative voltage if the system is only powered by a single supply voltage (such as from a laptop charger, for example). It is not possible to convert a positive voltage into a negative voltage with linear voltage regulators. Therefore, in some cases two power supplies are used, which increases the cost of the system.

Figure 12 shows a global overview of the power supply section of an audio DSP. This is used as a reference to show where which power supply topology can best be used. As shown in Figure 12 a combination of switching voltage regulators and linear voltage regulators will be used throughout the audio DSP. The digital domain will be powered by switching regulators since this domain is not very noise sensitive. The analog domain will be powered by low noise linear voltage regulators which in some cases will be pre-regulated by switching regulators.

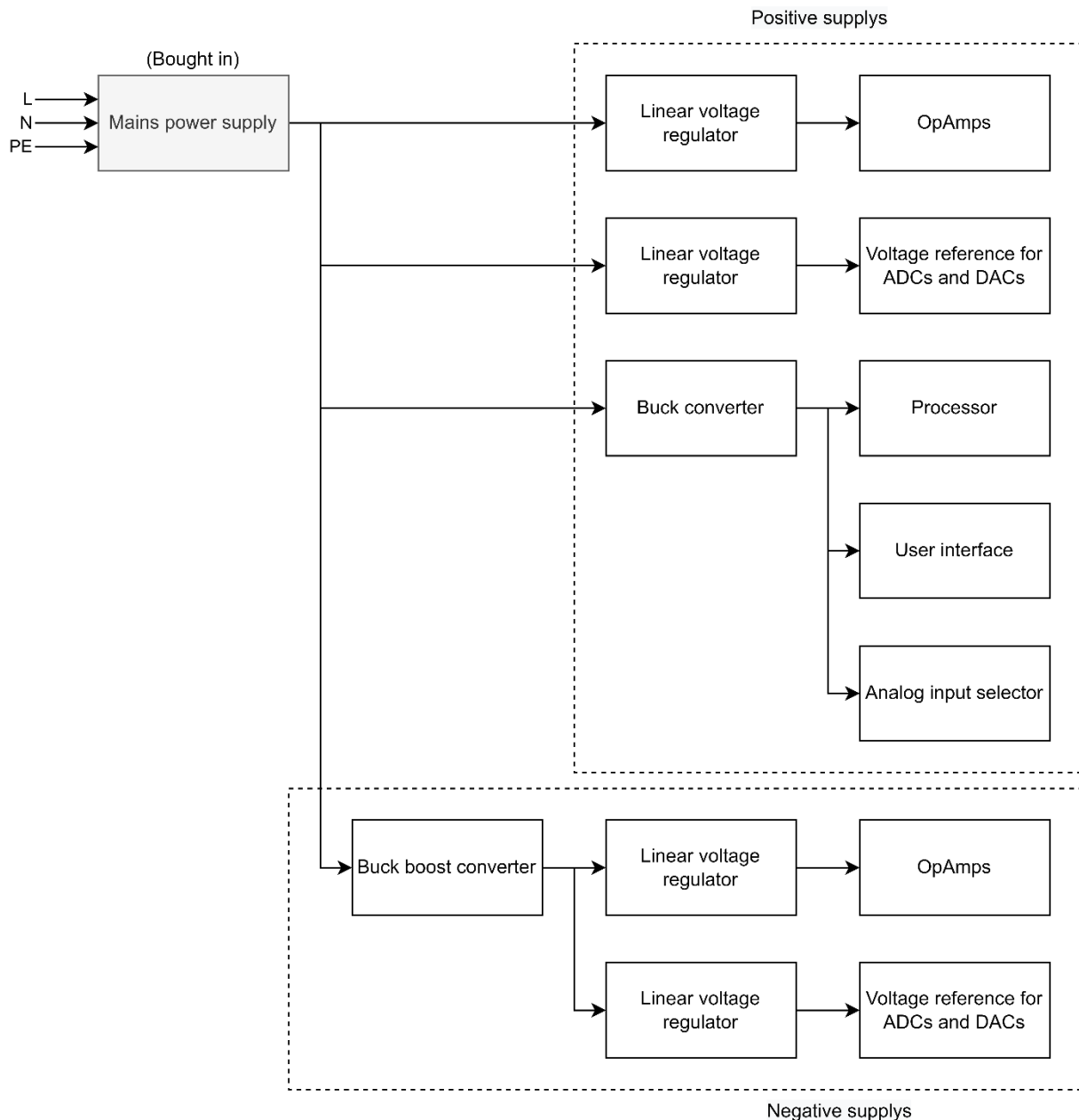


Figure 12: Global overview of the power supply section of an audio DSP.

Because switching regulators soon have a few millivolts of ripple voltage, it is necessary to thoroughly filter this incoming voltage because otherwise the noise of the switching regulators will end up with the noise sensitive analog section. This is because in datasheets of linear voltage regulators manufacturers often throw high PSRR (power supply rejection ratio) values, but this PSRR is actually frequency and temperature dependent. Thus, it is common for the PSRR to decrease with frequency as seen in Figure 13. Figure 14 shows a typical PSRR curve of a linear voltage regulator over temperature.

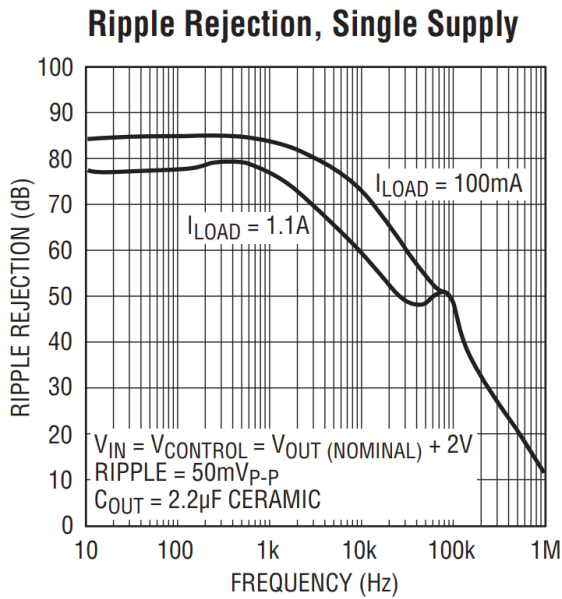


Figure 13: PSRR vs frequency of the LT3080 linear voltage regulator. Ref [10]

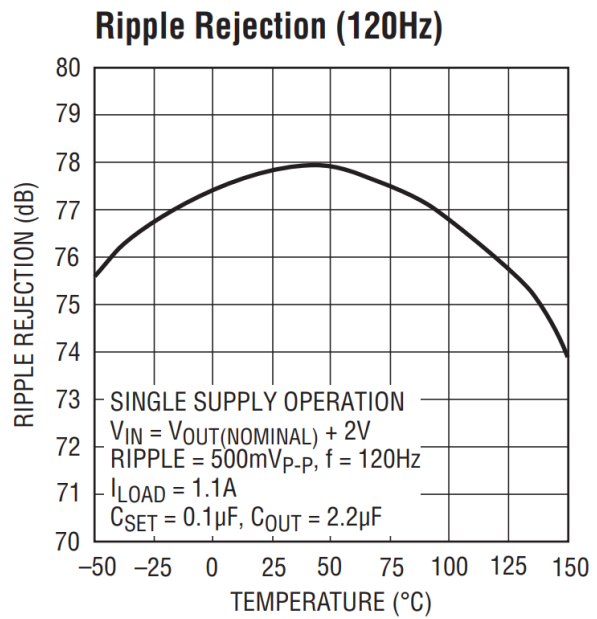


Figure 14: PSRR vs temperature of the LT3080 linear voltage regulator. Ref [10]

Since the linear voltage regulators as seen in Figure 12 will in all cases be powered by switching regulators which have high-frequency switching noise, it is important that the linear voltage regulators already get some pre-noise suppression. The simplest method for this is an RC filter. For a standard RC LPF, the cut-off frequency can be calculated using the formula below:

$$f_c = \frac{1}{2\pi RC}$$

To get a very low cutoff frequency and a high PSRR, either the resistance value R must become very large, so that the filter can no longer supply current because the output voltage would otherwise drop too far, or the capacitance C must become unreasonably large, making it very physically very large, and gets expensive. This behaviour can be easily improved by adding a transistor whose collector is connected to the (noisy) input voltage, the base is connected to the output of the RC network and the emitter forms the output. What this circuit effectively does is it multiplies the capacitance of the capacitor by the hFE of the transistor. This is because most of the current flows through the transistor and the capacitor therefore sees less ripple current. The cutoff frequency of the capacitance multiplier can therefore be calculated as follows:

$$f_c = \frac{1}{2\pi RC\beta}$$

Figure 15 shows the principle of the capacitance multiplier circuit. It is important to realize that this is not a voltage regulator because the output voltage simply fluctuates with the input voltage and there is always voltage across the collector-emitter junction of the transistor.

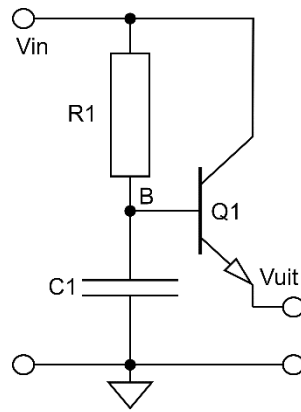


Figure 15: Capacitance multiplier principle

1.16 How do we ensure that the USB input of the processor is recognized by Windows as an audio device?

It is the intention that the USB input of the processor is recognized by Windows as an audio device so that it can be used for Spotify or YouTube, for example. For this, the processor must be able to sample the USB data and tell the computer what kind of device it is. A Teensy 4.0 microcontroller can be used for this, which has software libraries that make this implementation straight forward. This Teensy can transfer the USB data lossless to I2S data for the processor. The processor must be able to receive this I2S data anyway since that protocol is used for almost all audio ADCs and DACs. At the same time, the microcontroller could also be used to run the UI to send the new filter coefficients to the processor.

MAX3421E – USB controller with SPI interface <https://nl.mouser.com/datasheet/2/256/MAX3421E-1513231.pdf>

MCP2221A – USB to I²C converter with GPIO <https://www.farnell.com/datasheets/2182406.pdf>

MCP2210 - USB to SPI converter, only windows 7... <https://www.farnell.com/datasheets/1514122.pdf>

2 References

- [1] Mark, "What Frequency Response is Good for Speakers? Flat or EQ.," 24 November 2022. [Online]. Available: <https://makelifeclick.com/what-frequency-response-is-good-for-speakers/#:~:text=The%20best%20frequency%20response%20for,sound%20in%20the%20same%20range..> [Accessed 6 March 2023].
- [2] Planet Analog, "ADC Guide, Part 9: Total Harmonic Distortion," Planet Analog, 11 December 2012. [Online]. Available: [https://www.planetanalog.com/adc-guide-part-9-total-harmonic-distortion/#:~:text=Total%20harmonic%20distortion%20is%20defined,\) %20and%20\(2\)%20resp ectively.](https://www.planetanalog.com/adc-guide-part-9-total-harmonic-distortion/#:~:text=Total%20harmonic%20distortion%20is%20defined,) %20and%20(2)%20resp ectively.)
- [3] M. V. a. P. Scheelke, "How to avoid Aliasing Intermodulation Distortion (AID)," [Online]. Available: <https://www.digitalaudio.dk/media/4aid.pdf>.
- [4] François, "Blog Son-Video," 24 March 2021. [Online]. Available: <https://blog.son-video.com/en/2021/03/what-is-the-signal-to-noise-ratio-can-we-improve-it-and-if-so-how/#:~:text=It%20is%20generally%20considered%20that,or%20more%20for%20a%20preamp.> . [Accessed 2 March 2023].
- [5] A. J. Aude, "Audio Quality Measurement Primer," Intersil, 1998.
- [6] Analog Devices, "Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so," [Online]. Available: <https://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf>.
- [7] Tektronix, "What is quantization error and how does signal to noise relate to this?," [Online]. Available: <https://www.tek.com/en/support/faqs/what-quantization-error-and-how-does-signal-noise-relate#:~:text=Quantization%20error%20is%20the%20difference,nois e%20C%20to%20the%20sa mple%20signal..> [Accessed 6 March 2023].
- [8] "microchipdeveloper.com," Microchip Technology, Inc., 2021. [Online]. Available: <https://microchipdeveloper.com/>.
- [9] P. Horowitz, "The Art of Electronics," in *The Art of Electronics*, Cambridge University Press, 2015, p. 928.
- [1] "LT3080," [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/3080fc.pdf>.
- [1] T. Instruments, "Texas Instruments," [Online]. Available: <https://www.ti.com/data-converters/adc-circuit/overview.html> . [Accessed 2023].
- [1] T. Instruments, "Texas Instruments," [Online]. Available: https://www.ti.com/lit/an/sbaa337/sbaa337.pdf?ts=1678198615117&ref_url=https%253A%252F%252Fwww.google.com%252F . [Accessed 2023].
- [1] V. Saxena, "Boise State University," [Online]. Available: <https://www.eecis.udel.edu/~vsaxena/courses/ece614/Handouts/DAC%20Architectures.pdf> . [Accessed 2023].

- [1 “Analog Devices,” [Online]. Available: <https://www.analog.com/en/design-notes/dac-and-op-amp-generate-variable-negative-control-voltage.html> . [Accessed 2023].
- [1 B. Black, “Analog Devices,” [Online]. Available: <https://www.analog.com/en/analog-dialogue/articles/analog-to-digital-converter-architectures-and-choices.html>. [Accessed 2023].
- [1 W. Kester, “Analog Devices,” [Online]. Available: <https://www.analog.com/media/en/training-seminars/tutorials/mt-016.pdf>. [Accessed 2023].
- [1 D. Sheingold, “Analog Devices,” [Online]. Available: <https://www.analog.com/en/education/education-library/analog-digital-conversion-1986.html>. [Accessed 2023].
- [1 Texas Instruments, [Online]. Available: <https://www.ti.com/europe/downloads/Choose%20the%20right%20data%20converter%20for%20your%20application.pdf> . [Accessed 2023].