

EP. 2: **The Digital Audio Domain**

What really happens when sound enters your computer? In this episode, Aleyna Neivandt-Brown breaks down the analog-to-digital conversion process that makes modern recording possible. Learn how microphones, preamps, and A/D converters transform sound into digital waveforms, and gain a practical understanding of sample rate, bit depth, aliasing, dithering, and DAW workflows. Whether you're just beginning your audio engineering journey or looking to strengthen your technical foundation, this episode explains the science behind digital recording in an approachable, real-world context.

Written and Recorded by
Aleyna Neivandt-Brown

Produced by
Red Pin Productions

Original Release Date
August 26, 2026

Run Time
36:41

This transcript accompanies the official podcast episode
and has been lightly edited for the written format.
Listen to the Episode: https://www.buzzsprout.com/2639789/episodes/19709019

© 2026 Red Pin Productions, LLC. All Rights Reserved.

In This Episode

[00:00] **Introduction**
- Welcome
- Digital Audio Overview

[02:08] **Analog-to-Digital Conversion**
- Signal Flow Review
- Discrete Time Signals

[06:04] **Sample Rate**
- Nyquist-Shannon Sampling Theorem
- Aliasing
- 44.1 kHz Early Standardization
- 48 kHz for Multimedia
- Considerations for 96 kHz and 192 kHz

[18:55] **Bit Depth**
- Defining “Bit”
- Decibels and dBFS
- 16-bit and 24-bit Standards
- 32-bit Float
- Quantization Noise
- Dither

[30:55] **Digital Audio Workflow**
- Nondestructive Editing

[32:45] **DAWs**
- Pro Tools
- Ableton Live
- Logic Pro
- FL Studio
- Reaper
- Cubase
- And Others

[36:08] **Conclusion**
- Understanding A/D Conversion
- For More Info
- Thanks

Hi everyone, welcome to another episode of the Red Pin Podcast. I'm your host, Aleyna Neivandt-Brown, audio engineer, composer, performer, and professor, and the Founder of Red Pin Productions. This podcast serves to enlighten and hopefully entertain you with some science and technology, studio stories, and interviews with guests from across the audio industry. Today's episode is just me, walking you through an important technical and scientific concept, which is… **audio in the digital domain**.

We're diving into the scientific detail of the analog-to-digital conversion process with all its relevant gear and signal flow discussions, so you can better understand how the sounds in your recording booth show up as digital audio waveforms in your DAW and how to retain the best quality audio throughout the process. I'd consider this episode to be Intermediate level on the audio engineering spectrum though some advanced mathematics are mentioned. You'll need some prerequisite knowledge to fully engage with the concepts I'll be discussing, but overall, the topic of A/D conversion is one that I believe all audio engineers, amateur to professional, should take the time to deeply understand. So let's get to it.

[Red Pin Podcast Intro Music]

[2:08] **Analog-to-digital conversion** is the process by which electrical signals become the data that shows up as a waveform for you to record, edit, and mix in your Digital Audio Workstation. Most studios today are using DAWs for recording—meaning, all audio captured by their microphones that travel through their preamps and consoles, has to be converted into little digital blocks for Pro Tools or Logic to read and replicate in your edit window. Let's review some **signal flow** fundamentals so we're on the same page about what this electrical signal is and where the Analog-to-Digital Converter is on the signal flow chain.

[2:59] It starts with the microphone. Microphones are transducers, so they are responsible for converting the mechanical energy—sound waves—from the source—let's say, your singer—into an electrical signal that is usable by our recording gear. If you want an advanced look into the internal mechanics of microphones, check out Episode 5, where I talk all about transduction methods from moving coil to variable capacitance and more. Let's get back to our signal flow chain.

[3:37] After the microphone has done its job of converting the singer's sound into an electrical signal of the proportionate amplitudes and frequencies that she created, it travels through the studio into your preamp—either on your analog console or your outboard pre or maybe into your little

two-channel interface, and there we adjust the gain to bring the signal up to a usable level. But that signal isn't what goes into our computer. First, we have to change it into a language the computer understands—data. 0's and 1s. That's where the ADC comes in. ADC, by the way, is an acronym for Analog-to-Digital Converter. Imagine your ADC is like a translator. The singer has sung some words and created some sound waves, and your ADC is the interpreter who will take what the singer said and translate it into the language the computer can read. It does this by taking incredibly fast samples. Snapshots of the waveform at rates of 48,000 or 96,000 or even more per second. Your DAW will then use those snapshots, those points—we're talking discrete data points—to basically trace the waveform and show it to you in your edit window, as what looks like and sounds like the same sound wave that emanated from your singer's vocal cords, but it isn't. It's just data. In terms of signal processing, by the way, when we're talking about going from an analog waveform—an electrical signal—to a digitized waveform, we're going from **continuous time signals** to **discrete time signals**—point-based replicas of the sound wave using a sequence of values that correspond to particular instants in time—samples. And that word is very important in this discussion.

[*brief musical transition*]

[6:04] We measure the precision and accuracy of this conversion process in two main categories, Sample Rate and Bit Depth. **Sample rate** is the speed at which the ADC is taking snapshots. This sample rate must be at least 40,000 per second in the case of audio production and is usually even higher. This is for a specific reason, historically investigated by a Swedish physicist and electronics engineer named Harry Nyquist and later further proven by mathematician and computer scientist Claude Shannon. The **Nyquist-Shannon Sampling Theorem**, when it applies to A/D conversion, basically states that in order to digitize an analog waveform without aliasing, the sample rate must be at least two times the highest frequency. The **range of human hearing**—if you're unfamiliar—is 20 to 20,000 Hz. 20 Hz would be our low sub bass tones and 20,000 Hz—or 20 kHz—would be super high pitches, not quite dog whistle pitch, but close! And in fact, we lose that upper range as we age, so many people can't hear that high anymore, but they are still important frequencies when it comes to audio. In that upper range, any incoming signal is probably an overtone, a really high harmonic that is just part of the resonance of a sound or the room, or part of the noise makeup, so it's still important to sample that frequency accurately. According to the Nyquist Theorem, we have to sample at at least twice that rate. This to protect from **aliasing**. Aliasing might be an unfamiliar concept but you may already be familiar with the word alias. Alias kind of means disguised. And waveforms can be disguised if the sample rate is too slow to catch the actual shape of a

waveform. Otherwise it might take a snapshot of the peak of one cycle and then a snapshot of a different cycle entirely, meaning your DAW wouldn't recognize it as a 20,000 Hz frequency. It might mistake it for something lower or incoherent completely and just noise, known as aliasing distortion. So considering the highest we can hear is 20,000 Hz, we need to sample at at least two points on every one of those cycles, which would mean at a rate of at least 40,000 Hz, but some wiggle room on top of that is nice too. So it's no wonder that our **CD and streaming** sample rate is 44,100 Hz, or 44.1 kHz. That covers our bases when it comes to the Nyquist Theorem and aliasing concerns. But why 44.1 and not just 40?

[9:32] **44.1 kHz** became the standard back when Sony first released their PCM 1600 to encode audio onto video cassettes back in the 1970s. VCRs were the only thing developed at the time that were portable and could handle that kind of data storage, but it meant that the audio encoding would need to be compatible with the video rates and equipment they were already using—which wasn't necessarily divisible by an even 40,000, but rather a formula of video lines, fields, and samples. The process worked by encoding three samples of audio per active line of video per field… 44,100. [Math Proofs: Monochrome NTSC: 245 active lines/field × 60 fields/sec × 3 samples/line = 44,100 samples/sec. PAL: 294 active lines/field × 50 fields/sec × 3 samples/line = 44,100 samples/sec.] And since that became a standard, for compatibility's sake, this rate stayed the same even with the advent of the compact disc, and as short-lived as CDs were, the sample rate has stayed the same even still as the baseline standard for streaming distribution even today. Though, some platforms are beginning to stream lossless audio at higher sample rates like 48 kHz, and I imagine that trend will continue as our internet and devices continue to get faster and faster. Which is fantastic, because we're not usually recording in 44.1 in a professional studio environment, are we? We usually have to downsample to that rate. The current baseline sample rate in the **studio environment** is **48 kHz**. This is for a couple reasons both historical and compatibility-wise, ever since DAT was invented in the 1980s, Digital Audio Tape, which could record and play back audio at a higher sample rate than its predecessor. DAT was used largely throughout the 90s but like many technological inventions during the last 50-60 years, it all happens so fast and next thing you know, becomes obsolete when the next improvement takes hold of the market. So in 2005, Sony announced they were discontinuing it. We do still use 48 kHz sample rate because of its compatibility with video. 48,000 is divisible by the common frame rates—60 frames per second, 30 frames per second, 24 frames per second. Dividing 48,000 by any of those would result in a whole number. Meanwhile, dividing 44,100 by those video frame rates, 24fps for example, would result in a non-integer, right? 1,837.5. We can't have that! That ruins synchronization. We'd have skipped samples, dropped samples.

Incompatible. So we use 48 kHz as our sample rate in a lot of studio environments, which can be attributed to our **multimedia** production industry.

[12:53] But if I've just talked about how higher sample rates improve quality and reduce aliasing and how technology is moving so fast, wouldn't you expect that we're capable of recording at much higher rates? Well, you'd be right. Many studios can record at rates like **96 kHz** or even **192 kHz**. That's hard to even conceptualize! I don't know about you. But just because we can doesn't mean we should. There are several factors to consider when choosing these extremely high sample rates. By default, I'll tell you now, these are not necessary outside of very specific applications. Recording at this high a sample rate requires a ton of speed and space on your hard drive, so if you're also trying to run a bunch of plug-ins and do a lot of processing during playback, your computer can really struggle and your hard drive can fill up fast. Of course, our technology is improving so rapidly that this may not be an issue for you if you have the fastest computer on the market and a dedicated external SSD for your project. But for the average person, considering the difference in the listening experience for general music production, it's going to be inaudible to the vast majority of people, and you're going to have to downsample to 48k or 44.1, regardless, for distribution. So, it's not necessary to use 96 or 192. But here are a couple reasons you might. First, if you are not the final person in the chain—there are other people working on a project and exchanging files back and forth, and the session originated in a higher sample rate—you should maintain the same sample rate if you're adding or editing audio on that project. It should only be downsampled during mastering, and the mastering engineer will have specific tools to export the lossless versions and the distribution versions. But don't be the person who passes off a session that was in full 96 kHz, and you recorded 48 and downsampled everything else, because remember this: a DAW session can only operate in one sample rate. All the audio in a session has to match, otherwise some will play back at different speeds. Second reason you might want to use 96 or 192 kHz—if you're recording foley or other sound design elements and then you expect to pitchshift or timestretch this audio by a considerable amount—moving it down a couple octaves or stretching it out to two or three times its original length for a creative effect. Well, then you might want more sample points to work with. By nature, of course, pitch is determined by frequency, which is a time-based measurement—cycles per second. By stretching out a digital audio waveform, or lowering the pitch to have much fewer cycles per second, we don't magically make up data points where there weren't any. Take a recorded tone of 10,000 Hz. If we sampled at 44.1 kHz, we'd have about four samples per cycle of that sound—four points along each cycle of that waveform that we can use and manipulate in the digital audio realm. If we wanted to time stretch that tone to four times its original length—in other words

pitchshift that tone down by two octaves—the resulting frequency would be 2,500 Hz. Recording a tone of 2,500 Hz at 44.1, we'd get at least 17 sample points along each cycle of that waveform. But we didn't do that! We recorded a 10,000 Hz tone, and now we're working with a timestretched, pitchshifted version of it. We still only have four points of information. The difference between four and seventeen sample points can mean the DAW really struggles to trace those points into a discernable waveform shape resulting in audible distortion. But if you did this same example, but recorded at a sample rate of 192 kHz, even pitchshifting that tone down two octaves, you'd still have 19 sample points per cycle to work with—which is even more precise than recording an actual 2,500 Hz tone at 44.1k. Meaning, there would absolutely be no audible quality reduction or distortion even with extreme pitchshifting or timestretching. So those of you interested in post-production work like foley and sound design, using those kinds of tools and techniques, you can consider the benefits of those higher sample rates for your sessions, but keep in mind you need a **high performance computer** and a lot of **storage space** on your drive.

[*brief musical transition*]

[18:55] Alright we've talked a lot about sample rates. Let's zoom back out and refresh our memory as to what process we're really covering—Analog-to-Digital conversion. Remember I said, we measure the precision and accuracy of this conversion process in two main categories? Sample Rate and Bit Depth. So let's move on to the second measurement. **Bit depth**. The word "**bit**" is actually a portmanteau of the words "binary" and "digit." A bit is our smallest unit of measurement in computing—the most basic, singular choice between **0** and **1**. When applied to digital images, like the visual world, consider this—one bit, or just a bitonal image, could only have pixels that are either black or white. There'd be no nuance of grayscale. 2-bit would be you have two digits. They can still only be 0 or 1, but that does give you more combinations. For example, now you could have 0-0, 0-1, 1-0, or 1-1. If 0-0 is black and 1-1 is white, then 0-1 would be dark gray, and 1-0 would be light gray. Just by adding one more bit we suddenly have some nuance to our image. To continue on in this way, you can actually represent it with 2^n where that exponent is bit depth. Some examples to show you the power—pun intended—of exponential growth, 4-bit would be $2^4 = 16$. Now we'd have 16 different combinations or levels of nuance in our image from black to white of varying shades of gray. 8-bit would be $2^8 = 256$. 16-bit would be $2^{16} = 65{,}536$. So what does this mean when it comes to audio? Well, we used the example of black versus white and the grayscale in between. Now think of it as **silence** versus **maximum loudness** and all the **dynamics** in between. In 8-bit audio, we get only 48 decibels of dynamic range. 16-bit audio, we get

about 96 decibels of range. And in 24-bit audio, we get about 144 decibels of dynamic range—meaning there can be more nuance to the amplitude of each sample point during our analog-to-digital conversion. Quick mathematical clarification—the reason those decibel numbers didn't show quite as dramatic a leap as our 256 to 65,000, is because **decibels** are already measured on a **logarithmic scale**. Quick, abbreviated lesson on decibels—you can think of decibels and volume in threes and tens. For every 3 dB increase, the sound energy has doubled. For every 10 dB increase, the sound energy has increased 10x. But sound energy isn't the same as our perceived loudness. Ears—which are the O.G. transducers, converting the mechanical energy of all the sounds around us into electrical signals interpreted by our brain—are really good at sorting sound by what's important, both in terms of frequency and amplitude. Our ears can hear a range of about 120 decibels, where 0 is silence and 120 is painfully loud. In digital audio, we don't have the same scale. The sound isn't naturally occurring—it's a matter of data, which, as we've discussed, can be 0 or 1, and the combinations of bits is what determines our levels of range within that. So, in the digital domain, we'll refer to dynamic range as it relates to Full Scale, or **dBFS**, decibels relative to Full Scale. And we actually make 0 the ceiling, and then use negative numbers below that. So it's common to aim for audio to peak around -3 dB, etc. Okay, decibel lesson over. Now back to the bit depth conversation.

[24:20] DAWs allow you to choose the bit depth of your session—typically 16-bit, 24-bit, or 32-bit float. 16-bit is the CD-standard and has been the standard for streaming distribution for awhile, but as I said before, with internet and device speeds constantly improving, we are seeing streaming platforms allow for 24-bit audio. 24-bit is typically the bit depth used in professional studio settings. So why use 32-bit float, and what does float mean? **32-bit float** allows for a kind of insane dynamic range—like over 1500 dB—which means digital clipping is impossible, and the noise floor is so low it won't even interfere with a whisper. How? "Float" refers to a floating decimal point, but there's also a fundamental difference in reading a 32-bit word compared to our 16-bit and 24-bit options. Rather than fixed decimal point 16-bit or 24-bit, which creates an integer value—whole numbers, in 32-bit float we're actually going to use **scientific notation**—decimals and exponents, right? Think: 1.46×10^7. In our 32 bits, the first bit indicates a positive or negative value. The next eight bits indicate the exponent, and the last 23 bits indicate the mantissa—the significand, the meaningful numbers of our scientific notation. This allows for such a wider and more precise range of numbers. But we're not done yet! 32-bit float also does not use the same ceiling as 16-bit or 24-bit, which use 0. In 32-bit float, our ceiling extends over 700 dBFS in both directions, positive and negative. Remember when I said digital clipping is impossible and the noise floor is so low it won't even interfere with a whisper? There's your mathematical proof.

[26:52] Okay enough math, now I'll pose the same question as we did with regard to sample rate. If higher bit depth means higher level of precision, do we need to use the highest possible for our audio to sound good? No. Just like with 96 and 192 kHz sample rates, 32-bit float is really only used in specialized settings, like in sound design environments where you're recording something with massive and sudden dynamic contrasts. And again, it requires high CPU performance and results in huge file sizes—33% larger than 24-bit audio files. So you need a lot of storage space for your session folder. But can we hear a difference? Now that question is a bit more interesting than the sample rate side of this conversation. When it comes to bit depth, you can. But not in a, "oh this 8-bit audio sounds like trash, and this 24-bit audio is pristine," there's a very specific element between the two that you can hear… noise. Analog-to-Digital converters in this day and age are very good at turning analog signals into discrete digital steps using delta-sigma or successive-approximation architectures, which includes noise-shaping loops. But noise is a given during this process. What is this noise and why does it happen? It's called **quantization noise**, and it occurs when an analog signal's sample has to be rounded to the nearest digital step. Some rounding errors occur, are measured and filtered, and then pushed into higher frequencies where they shouldn't interfere much with the main audio content. It shouldn't be noticeable, but if you're recording quiet, high frequency content, this quantization noise could pose an issue. This kind of noise can also occur during mastering. When you're taking your 24-bit audio and bouncing it for streaming distribution in 16-bit, the same quantization errors can occur when rounding all those points from 24 digits of data down to 16—cutting off 8 bits of precision—resulting in some digital distortion which is audible as noise, especially during a soft moment like the end of a reverb tail. That's where **dither** comes in. You can apply dither at the end of your mastering chain to apply some random noise. I know, it sounds counterintuitive. Fight noise with more noise? But doing this softens the truncation and smooths out the new 16-bit dynamic range. There are different algorithms to apply the noise, including Probability Density Functions—either rectangular or triangular—and noise-shaped algorithms that throw the noise into the outermost frequencies so it's less perceived. Dithering is also done in visual bit-depth reduction, but it's harder to describe that on a podcast, so just imagine the black versus white versus grayscale conversation from earlier and how lots of little black and white specs can actually give the impression of grayscale. That's what I mean by adding noise to smooth out the noise.

[*brief musical transition*]

[30:55] So now we've covered the two important specifications of Analog-to-Digital conversion: sample rate and bit depth. But what can you do with audio in the digital domain? Well, once that

waveform is in your DAW, you have the power of **nondestructive editing**. Unlike the physical medium of tape, you can splice and crossfade two audio clips, and if you don't like it, just press undo. You can cut, paste, duplicate, loop, trim, turn up, turn down, pitchshift, you name it, and you can revert back to the original audio file if you decide to go a different direction. Because it's all just data. In fact, that waveform itself can be manipulated. You can zoom in and use the Pencil tool to literally redraw the waveform, like to get rid of clicks or pops. You can even quantize your audio to snap transients to the grid—just like with MIDI—using Elastic Audio or Beat Detective in Pro Tools. In the digital domain, we have so much more flexibility, tools, speed, and forgiveness. Mistakes don't cost you physical tape or studio time to re-record something. If you're making a lot of edits or working your way through a project with some experimentation, I'd recommend that you save your session file with a new name during different stages of your process, so that you can always go back to before any wrong turns. Another luxury of working in digital audio.

[32:45] Who are some of the big players in **digital audio workstations**? Well, we have Pro Tools by Avid, often referred to as the industry-standard. It's compatible across Windows and Mac. It was early on the scene and provided the best recording workflow, and for the most part has kept up with modern electronic production, too. But if you're really into electronic production and want to DJ or play and loop in a live setting, I'd recommend Ableton Live. I am primarily a studio engineer, so I focus on recording, mixing, and mastering. So my primary DAW is Pro Tools Ultimate, and I'm an Avid-certified Specialist and Instructor. That being said, when I'm producing an electronic track—for myself or for my clients—I actually like to start my brainstorming in Logic Pro. Logic is made by Apple and is a Mac-only software, which is why it isn't quite as ubiquitous in the professional studio space, but it's very intuitive, and for MIDI production, it definitely thrives. All of these are professional DAWs, but if you're just getting started and you need something more budget-friendly, GarageBand comes stock on Apple products. Reaper is free to download and start using with full functionality, though they request after the trial period that you pay to continue using it. FL Studio, Fruity Loops, is a pretty capable DAW, but just beware, a lot of choices are made for you to streamline production and make it more accessible, but that means you're not learning as much under the hood, but it does excel in the beat-making space. Just keep in mind, Ableton and FL use very different workflows than Pro Tools, so the skills may not transfer to the studio setting as well. Cubase, made by Steinberg, isn't as popular in the studio environment but certainly capable and great in the MIDI and media scoring realm. There's also Studio One and Adobe Audition, neither of which I use personally so I can't speak on those. Universal Audio developed their own DAW called LUNA, which I also haven't used much. And throwing it back, I first learned some digital music

synthesis in a class in my undergrad that used Digital Performer, which is by MOTU, Mark of the Unicorn. They also make interfaces. I don't personally know of anyone still using Digital Performer, and I remember it having a pretty steep learning curve. I gravitated towards Cubase for film scoring early on, and then found myself learning and getting certified in Pro Tools almost ten years ago now, and it's been in every studio I've ever worked at or stepped foot in since then. This episode isn't even sponsored by Avid, that's just my experience.

[Red Pin Podcast Outro Music begins in the background]

[36:08] Anyway, thanks for sticking around through all this technical talk about audio in the digital domain. I hope you feel more grounded in your understanding of the analog-to-digital conversion process and how it applies to your own workflow. For more information, visit redpinproductions.com. I'm Aleyna Neivandt-Brown. Thanks for listening.

References and Further Reading

SoundGuys. "Audio Bit Depth Explained." *SoundGuys*. Accessed June 2026.
https://www.soundguys.com/audio-bit-depth-explained-23706/

Sound Devices. "32-Bit Float Files Explained." *Sound Devices*. Accessed June 2026.
https://www.sounddevices.com/32-bit-float-files-explained/

Pro Tools | Avid
https://www.avid.com/pro-tools

Ableton Live
https://www.ableton.com/en/

Logic Pro | Apple
https://www.apple.com/logic-pro/

GarageBand | Apple
https://www.apple.com/mac/garageband/

Reaper | Cockos, Inc.
https://www.reaper.fm

FL Studio | Image Line
https://www.image-line.com

Cubase | Steinberg
https://www.steinberg.net/cubase/

Suggested Citation

Neivandt-Brown, Aleyna. "Ep. 2: The Digital Audio Domain." *Red Pin Podcast*. Red Pin Productions, LLC. August 26, 2026.

Support the Red Pin Podcast

Become a donor on Patreon at a monthly membership level that works for you. Visit https://www.patreon.com/cw/RedPinProductions to learn more.