Digital Performance Data: MIDI Is Not Digital Audio

A MIDI file contains no audio data; that is, the MIDI file format contains no digital audio, only **digital performance data**. This performance data is played back into the synthesizer by the computer, using the MIDI hardware (interface, cables, and ports) that connects the computer and synthesizer together with an **in** and **out** cable, so that your computer and synthesizers can talk to each other. As the computer is playing back your existing track using the synth **in port**, it is recording the next track, which you are composing as you listen to the computer play your previous MIDI tracks.

The computer is therefore recording the data coming from the out port on your synthesizer at the same time it is playing your existing tracks into the in port on your synthesizer. Fortunately, computers are able to process MIDI data rapidly!

There is also a MIDI **through cable**, so that more than one synthesizer can be connected simultaneously. My MIDI synthesis setup includes the Yamaha TX-802, which is eight DX-7 synths in one rack mount, the Roland D-50 rack mount, and the Korg Z1. My setup only needs one keyboard synthesizer due to this through port, because performance data that I played on a Korg Z1 routes into the TX-802 and D-50 for rack-mount synth module playback.

MIDI records the piano keys pressed on a synthesizer keyboard or a sampler keyboard. It also records the keypress duration, the amount of pressure the key was pressed with (the **aftertouch**), and similar playback performance nuances.

It is important to note a third type of keyboard, called a **controller**, which is used only to generate MIDI performance data. It looks like a synthesizer or sampler keyboard, but it requires a rack-mount synthesizer or sampler, or digital audio software, to trigger samples or synthesize the sound waves that the MIDI performance data would trigger.

Audio Synthesis: Synthesizers Create Sound Waves

MIDI performance is silent. As you have seen with Rosegarden, it records your performance data, while the digital audio waveforms created by that performance are actually made with your keyboard synthesizer. Your synthesizer, or "synth," generates artificial or "synthesized" audio tones using the MIDI performance data to specify how to create your digital audio (synthesized) sound waves. A **sampling keyboard** plays back a digital audio sample (pre-recorded waveform) based on the MIDI performance data. It provides an even more aural-realistic (like photo-realistic imagery) performance, because each note on the instrument is sampled, or recorded, whether it is a piano, guitar, bass, fiddle, banjo, horn, flute, oboe, or drums.

When MIDI files are played back through the synthesizer, or through a sampler, it replicates the exact performance of the performer or the composer; even though that person is no longer playing the performance track, the computer is playing it back.

Let's discuss the way MIDI data is used in MIDI sequencer software. You play an instrument track, record the instrument performance for your music composition using MIDI data, and the MIDI sequencer then plays the performance while you play a second instrument track—alongside the first instrument's performance track.

While the computer is handling the MIDI performance data (recording and playback), the controller keyboard is handling the performance data generation and the rack-mount synthesizer or sampler is processing the performance playback MIDI data, which it is receiving from the MIDI sequencer software.

MIDI enables songwriters to assemble complex musical compositions and to refine them into any number of arrangements using only a personal computer, a synthesizer, a sampler, or a digital audio production software package. Composing or arranging in your home on a workstation costs less money than hiring a recording studio full of studio musicians! This is why I had you download and install Rosegarden—so that you can play around with MIDI, which makes the concept easier to understand.

MIDI Platform Support: Android, HTML5, and Java

All the open source platforms support MIDI files, including Android, HTML5, Java, and JavaFX. I expect closed platforms such as iOS and Windows to also support MIDI performance data playback. It is important to note, however, that **MIDI playback hardware**, such as a synthesizer or a sample library, must be in place for MIDI playback to succeed.

MIDI playback is also possible when a **MIDI playback software** capability that mimics MIDI playback hardware (audio synthesis or sampler) is installed. For this reason, MIDI is best utilized in an audio production and music composition scenario; although not so much in a content delivery scenario.

Data footprint optimization can be significant when using MIDI. These data-heavy waveforms can be stored on the "client side" and "rendered" by extremely data-compact MIDI performance data as needed by your musical composition. This is significant because the same note needs to be recorded only once (see Chapter 3), but can be used in a composition a million times by using the same data in memory. The more complex programming and optimization principles are covered at the end of this book.

Android OS supports the playback of MIDI files, but it does not implement a MIDI class for their creation. I hope that this changes in the future, but as it sits right now, it is not an easy job to code MIDI sequencers in Android Studio; however, this is being discussed in coding forums.

JavaFX also supports the playback of MIDI file formats, as does HTML5, so devices such as iTVs, tablets, e-readers, smartphones, and smartwatches support MIDI if you need to utilize it for your multimedia production projects.

Summary

In this chapter, I made sure that you had a MIDI sequencing and scoring software package installed and ready to master, just in case you want to be a music composer and an arranger, in addition to being a digital audio editor and engineer. You looked at the history of bridging analog audio with digital audio, and I discussed MIDI, MIDI sequencers, synthesizer keyboards, MIDI controllers, and sampler keyboards.

You learned about how MIDI works using only performance data, how controllers can generate that data, and how synthesizers or samplers can generate sound waves (covered in Chapter 1) using this MIDI performance data.

In the next chapter, you look at the concept of sampling, or taking data samples of the sound wave to convert it from analog waveforms into a digital audio data format.

CHAPTER 3

The Reproduction of Digital Audio: Data Sampling

Now that you know the general history of digital audio, it is time to get into digital audio sampling (essentially the same as digital audio recording), which needs to be done before any digital audio editing can be done. This chapter covers the concepts behind "digitizing" analog audio—or turning it into digital audio—by defining which **sampling frequency** is utilized during the digitization process and which **sample resolution** is used to store individual samples of analog audio waveforms as digital audio data.

In addition to digital audio sampling concepts, such as sampling frequency and sample resolution, you also look at standard sampling frequencies and sample resolutions, which are used in the digital audio industry, and how to sample, or record, digital audio by using Audacity 2.1.1 for Windows.

Data Sampling: Resolution and Frequency

In this chapter, I cover the process of **digital audio data sampling**. You will see how the transition can be made between analog audio and digital audio using the process of sampling, which you are already familiar with if you do sound design and music synthesis.

Sampling is even discussed in the news, as recording artists are sometimes sued for taking little "riffs" or "snippets" of another artist's song and sampling it in their songs. For this reason, be very careful with what you are sampling; make sure that it is an original analog audio waveform or that you have permission to use it! I want to make sure that you have a firm understanding of the digital audio new media assets that you will eventually create, optimize, and "render" via open source APIs, such as Java and JavaFX, or the Android Studio digital audio playback classes, SoundPool and MediaPlayer.

Sampling Digital Audio: Taking a Data Sample

The process of turning analog audio (sound waves) into digital audio data is called **sampling**. If you work in the music industry, you have probably heard about a type of keyboard (or rack-mount equipment) called a "sampler." Specifically, sampling is the

Electronic supplementary material The online version of this chapter (doi:10.1007/978-1-4842-1648-4_3) contains supplementary material, which is available to authorized users.

W. Jackson, *Digital Audio Editing Fundamentals*, DOI 10.1007/978-1-4842-1648-4_3