Everything is numbers

Sampling sound

Ever wondered why some music/sound that you hear in games or on the Internet sounds better than others? You might even notice that if you listen to something on the radio, that the same song sounds better when you listen on CD or stream from the Internet. This all has to do with the amount of data being used to store the sound, whether that's on a CD, your computer or the Internet.

Sound waves

Did you know that sound travels in waves? You can investigate sound waves (and other types of waves) using an oscilloscope, which gives us an easy to understand view of how the sound moves around. You can experiment a little with sound using the SoundLab tool:

Seeing with sound[1]

Notice how when you 'ping' a sound, it is displayed on the scope as a wave? How does changing each of the following alter the wave?

1. Increasing the frequency?
2. Increasing the wavelength?
3. Increasing the amplitude?

The shape and size of the wave therefore has a significant effect on the sound that you hear. A high frequency wave (which has a short wavelength) has a higher pitch than a wave with a lower frequency, and a sound with a high amplitude is louder than one with a low amplitude.

If we want to represent this in digital format on a computer, we need to store all of this data. For music, which is made up of a lot of different sounds that all have their own sound wave, it can get really complicated!

So, how do we store this information so that we can reproduce sound when we need to access it?

Analogue to digital converter (ADC)

We need to somehow convert our wave – which is an analogue signal – into a digital signal that is represented as a series of numeric values (which can then be stored as binary data).

The way we do this is dependent on two things:

- our sample rate
- our bit resolution.

[1] http://www.scootle.edu.au/ec/viewing/L1301/index.html

Take a look at the diagram below:

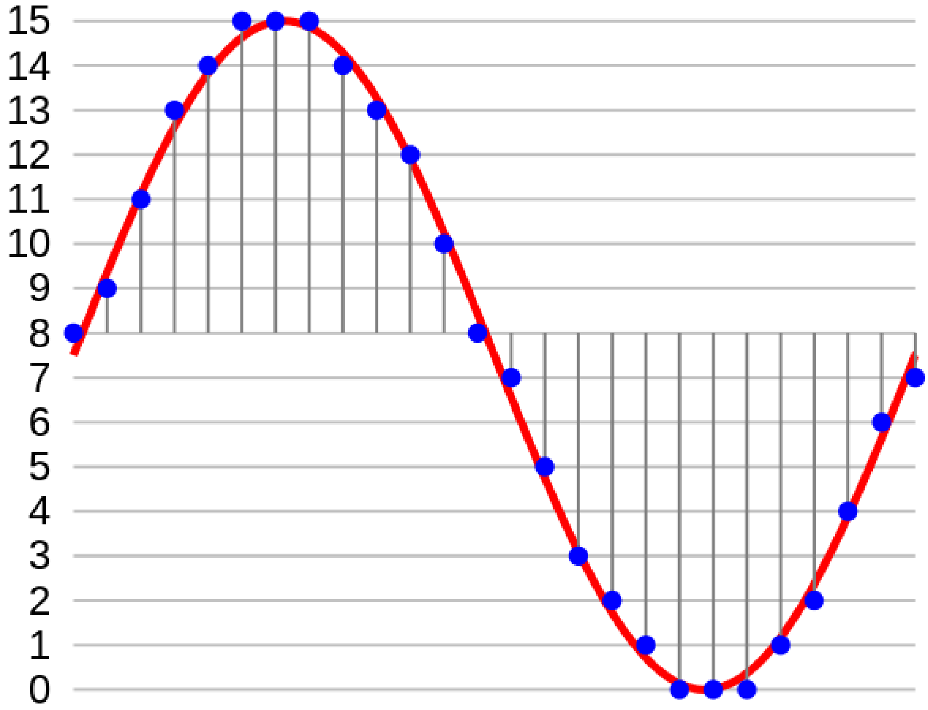


Image credit: https://commons.wikimedia.org/wiki/File:Pcm.svg [2]

The red line is the wave that represents the audio data we are attempting to store. Notice that there are 16 possible values that we can assign to the wave (from 0 through to 15). These 16 values can be represented in 4 bits (since 4 bits allows us to store 16 unique values), so we say that the data we are storing would have a resolution of 4-bits.

We then identify our sample rate – how often we will be checking the wave and storing its value.The blue dots have been placed on the wave at equal intervals, as shown by the grey, vertical lines.

- If we reduce the space between each line, then we would be storing more values and this means we would be increasing the sample rate (i.e. taking more samples). This would give us values that match more closely to the original wave.
- A lower sampling rate (with wider spaced lines) would be further apart, and result in a less exact match to the original wave.

You'll also notice that some of the blue dots aren't on the wave, but instead are placed on the value closest to it. This is because in our 4-bit resolution we can only use 16 values, so where the wave doesn't line up exactly, we need to approximate to the closest value.

If we increase our resolution (to 16-bits, for example), we would be able to store more values, and we would end up with a wave much closer to the original.

[2] https://commons.wikimedia.org/wiki/File:Pcm.svg

This wave, would thus be stored using the following (decimal) values:

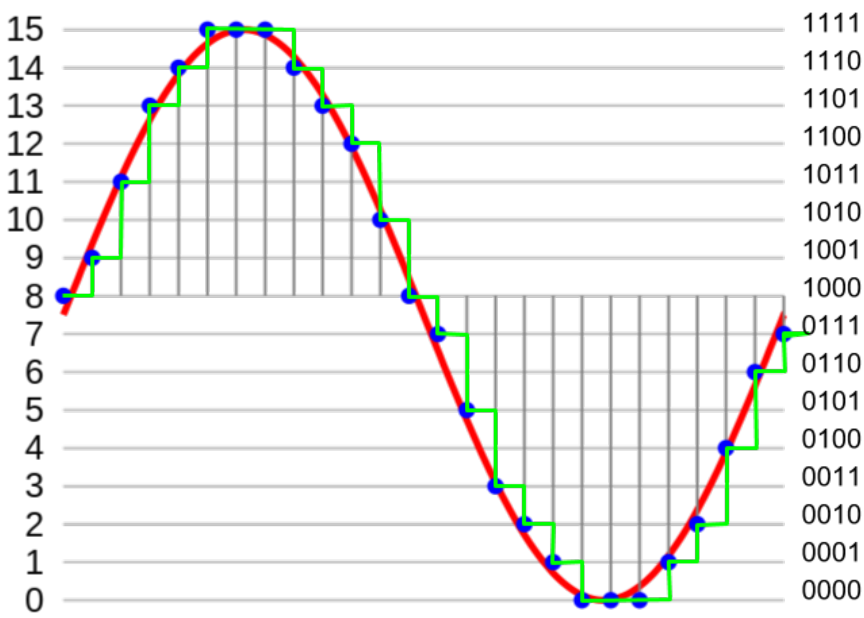
8, 9, 11, 13, 14, 15, 15, 15, 14, 13, 12, 10, 8, 7, 5, 3, 2, 1, 0, 0, 0, 1, 2, 4, 6, 7

Or, in binary representation:

1000, 1001, 1011, 1101, 1110, 1111, 1111, 1111, 1110, 1101, 1100, 1010, 1000, 0111, 0101, 0011, 0010, 0001, 0000, 0000, 0000, 0001, 0010, 0100, 0110, 0111

If we then count up the number of bits used to store this wave, we would have 26 values x 4 bits = 104 bits = 13 bytes.

If we were to then reproduce the wave, it would end up looking like the green line on the image below:



The closer the match to the original red wave, the better the quality of the audio that is reproduced.

Your turn

Now it's your turn to experiment with this idea. Using the same wave, what happens when you do each of the following?

1. Double the sample rate, but keep the bit resolution at 4-bits
2. Increase the bit resolution to 5-bits, but use the original sample rate
3. Halve the sample rate and use a 5-bit resolution

The original data points have been left on the image to help you compare the new waves with the provided one. You will need to mark the new data points (use a different colour) and draw a new reproduced wave in each case.

Templates that will help you do each of these things have been provided below.

Once you've drawn the new waves, answer the discussion questions at the end of this worksheet.

Double sample rate: 4-bit resolution

Stored values

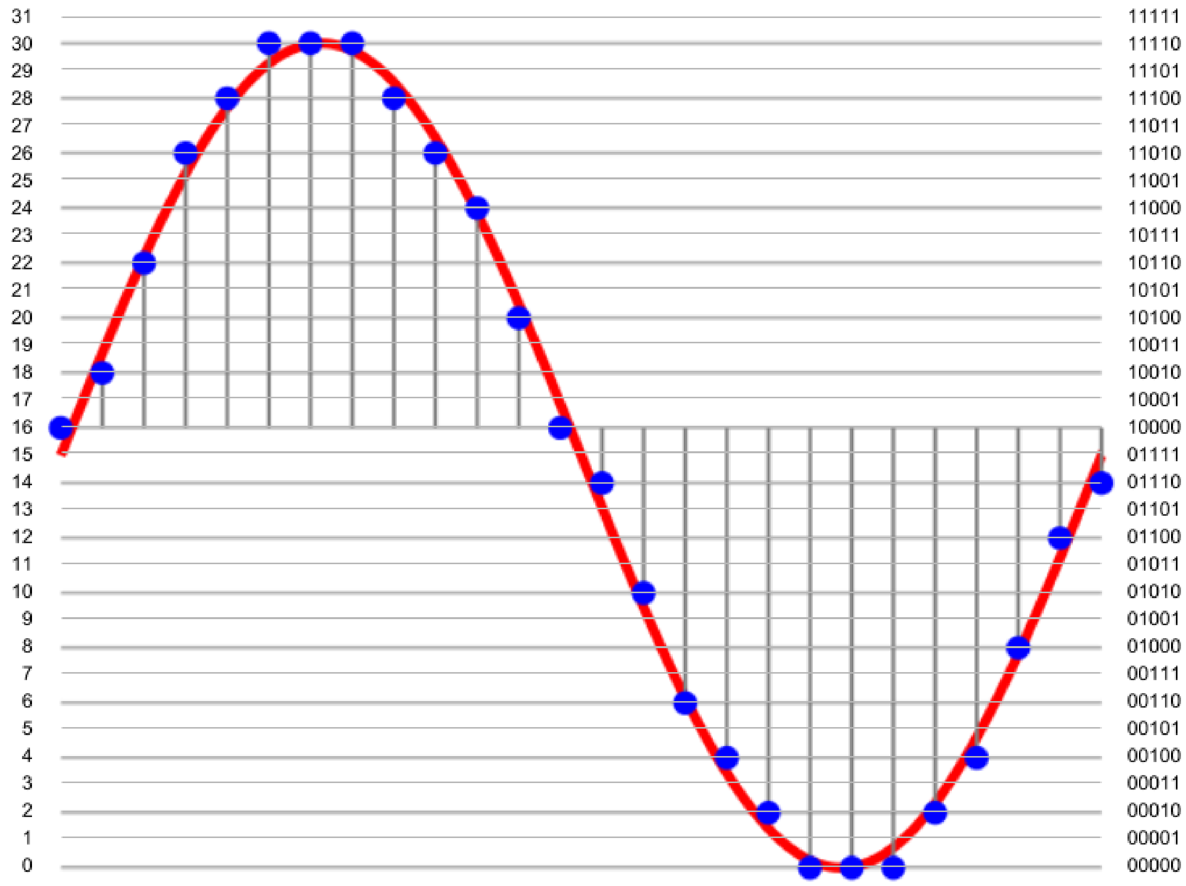
Decimal

Binary

Total size (bytes)

DIGITAL TECHNOLOGIES HUB

Original sample rate: 5-bit resolution



Stored values

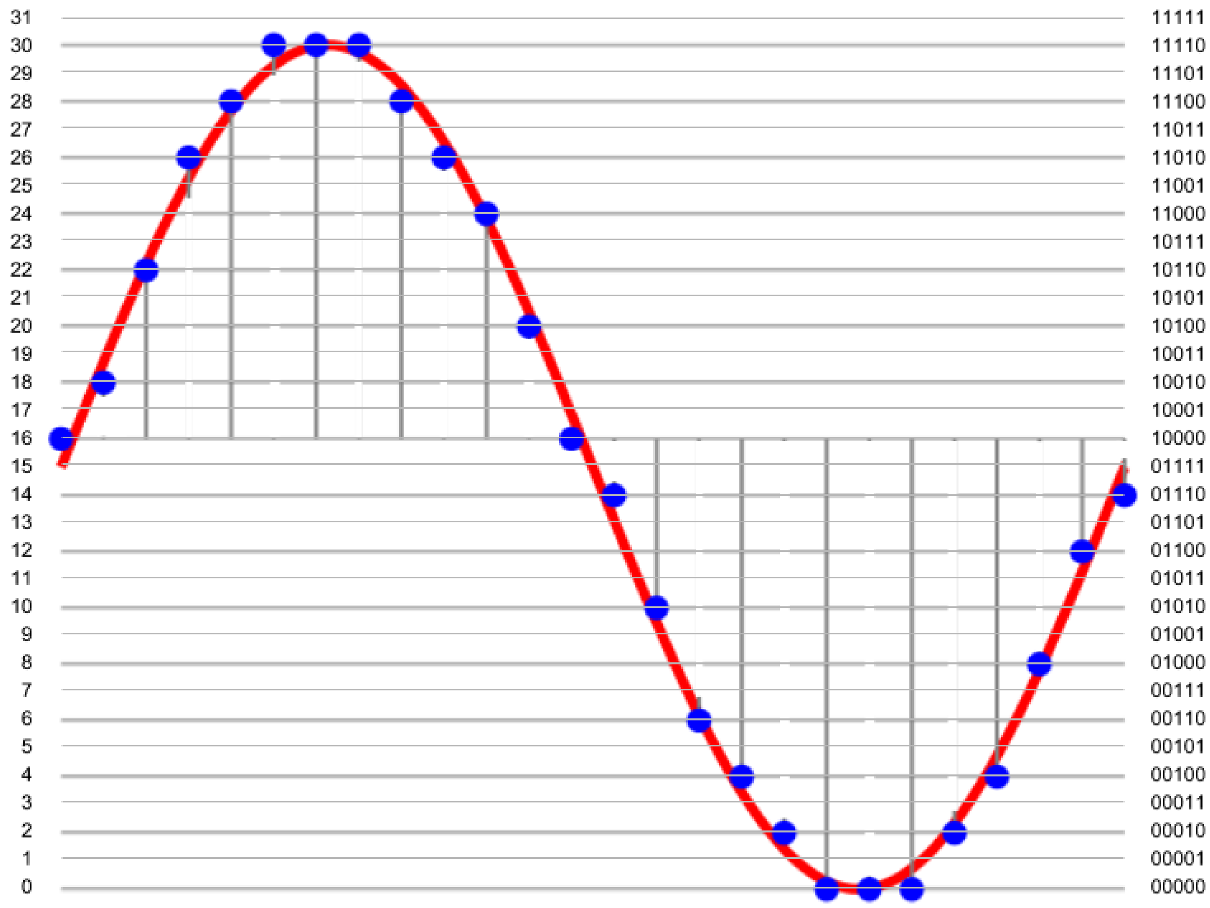
Decimal

Binary

Total size (bytes)

DIGITAL TECHNOLOGIES HUB

Halved sample rate: 5-bit resolution



Stored values

Decimal

Binary

Total size (bytes)

DIGITAL TECHNOLOGIES HUB

Discussion questions

Compare the different reconstructed waves you have drawn for each of the sample rates and bit resolutions.

1. Which factor has the most significant impact on the quality of the audio - the sample rate or the bit resolution?
2. Which factor has the most significant impact on the size of the file?
3. What is the relationship between the size of the data and the quality of the reproduced audio?
4. Using what you have learned throughout this activity, make a conclusion about the accuracy of the following statement, using evidence:

   In most cases, there would be very little noticeable difference between an audio file with a sample rate of 44kHz and a 24-bit resolution, and an audio file with a sample rate of 44kHz and a 16-bit resolution. When saving music to play via the Internet, you can sacrifice some of the quality to ensure the data transfer rate is fast enough for streaming, yet still maintain audio quality good enough for listening.