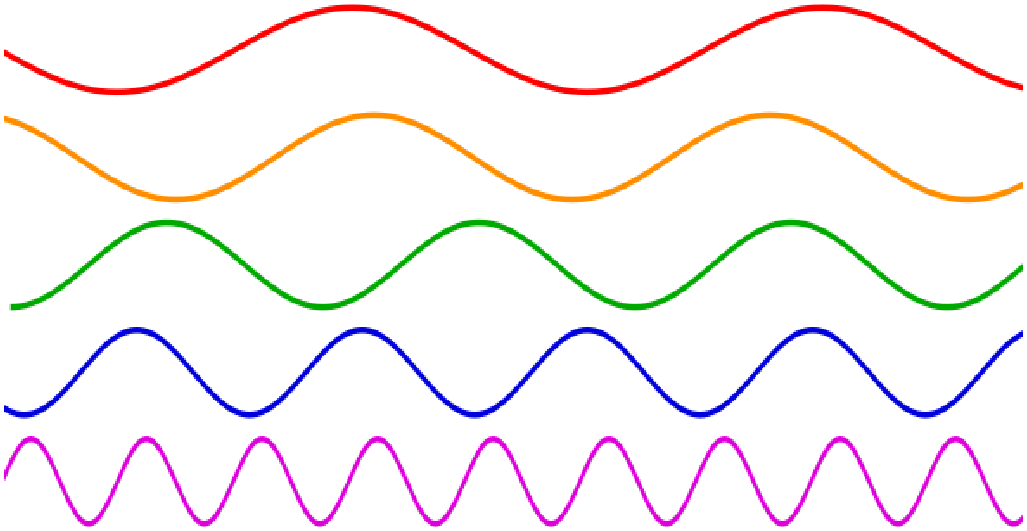
Sound and the Codec Output

On the DE2

Sound

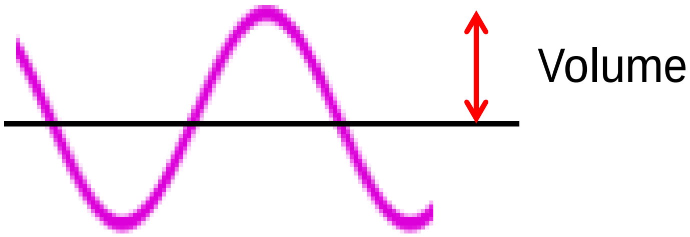
Sound is transmitted in air through compressions and rarefactions (stretches) of the air. This can be represented as a wave, symmetric around the central, “idle” value:



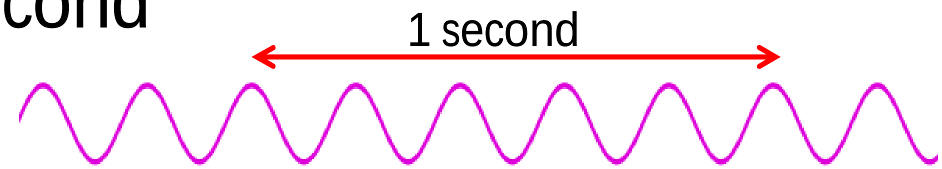
As shown, the waves of different sounds are not the same

Volume & Frequency

The volume is represented by the range of the wave up and down, that is its height above and below the "centre".



How many waves per second is the frequency.



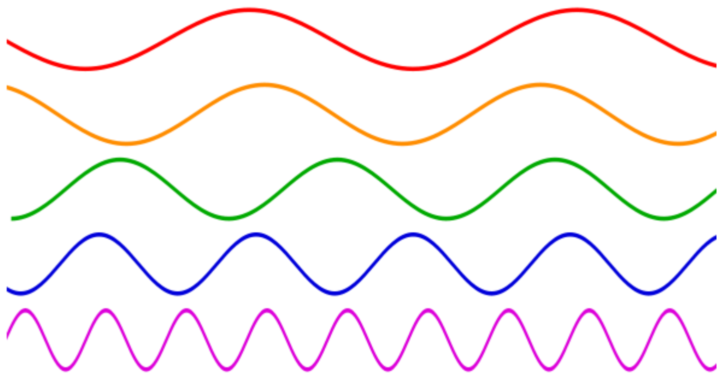
Frequency = 5 waves per second

Waves

The more waves per second, the higher the frequency and the higher the sound.

Hertz (Hz) is the same

as waves per second.



Humans hear sounds from roughly 20 Hz (or waves per second) to 20,000 Hz or 20 KHz; human voice range is about 80 Hz to 1100 Hz, with musical instruments going beyond this.

Music

Most people are aware of an octave. Two notes or sound waves are an octave apart if the higher note is double the frequency (or Hz) of the lower note.

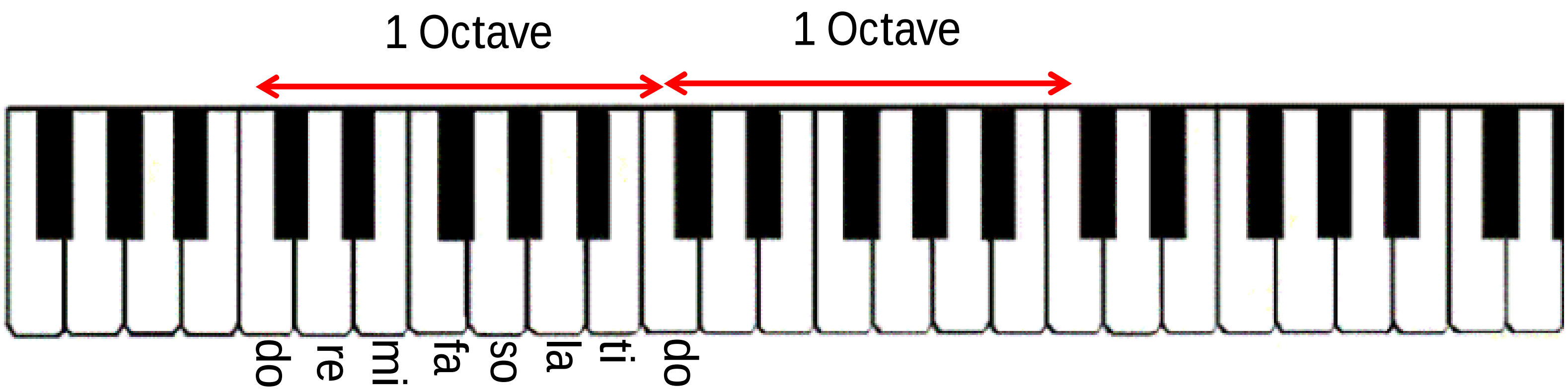
The two “do”s in the chromatic scale:
do re mi fa so la ti do

are an octave apart, thus double the frequency.

(more next slide)

Octaves cont

Piano keyboards have repeating patterns over an octave



Each octave the frequency doubles.

For our purposes, just note that the difference between say 100Hz and 200Hz will sound the same to humans as the difference between 1KHz and 2KHz.

The Codec

We are only going to discuss one “channel” of the codec which is the thing to output sounds. (We also will ignore sound *input*.) There are two channels for stereo sound. We will assume you will treat both channels the same.

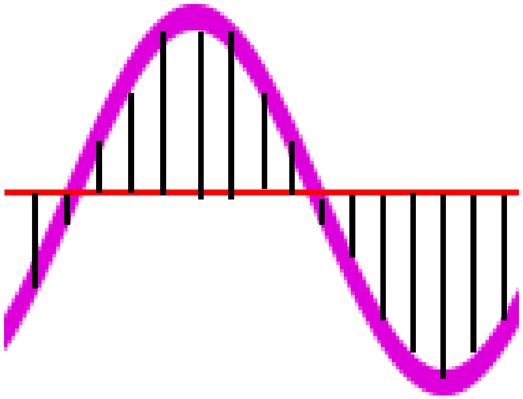
The codec does NOT put out a sound. It puts out an instantaneous sample of how rarified the air should be at any specific time.

Codec Samples

The codec expects 44,000 samples every second that it can put out to the speaker. The actual values are signed values, up to +/- 32,767 (ie 16 bits of significance) representing the deviation from the “neutral” centre point. The magnitude represents the volume.

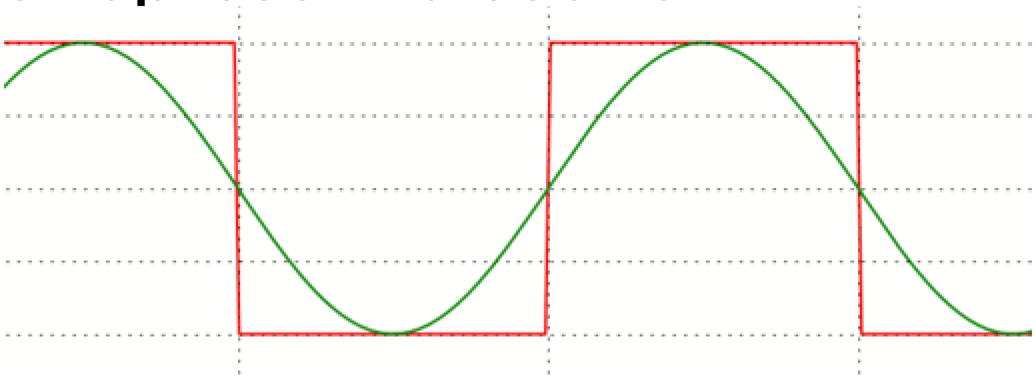
Here the black lines

show the samples.



Sounds - easier

To make your job easier, use square waves (the red in the figure) – this means you put out a single value for a certain length of time, then its negative, to represent a sound.



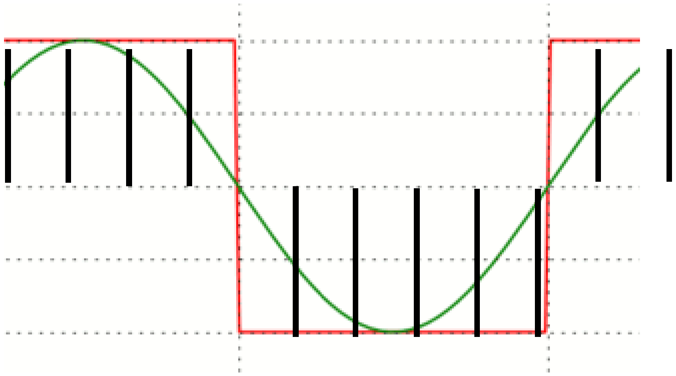
The next slide shows the sampling

<<Slightly harder: Use triangular waves>>

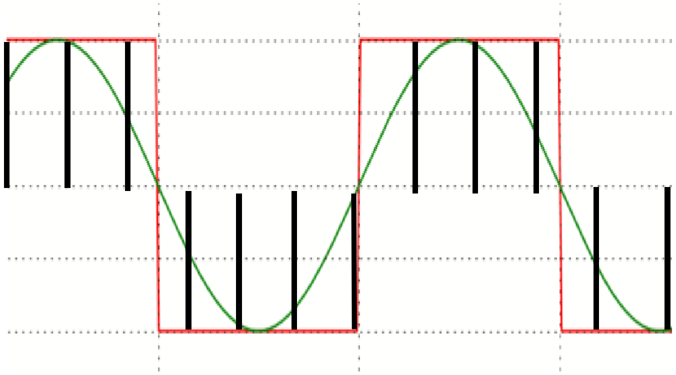
Samples vary with frequency

Higher sounds will change faster

Low sound

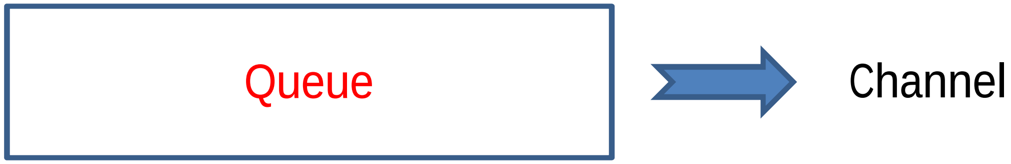


High sound



Updating the Channel

To make things easier, each channel has a queue (or “FIFO”) in front of it. The channel, the speaker system, will grab values from the queue 44,000 times a second.

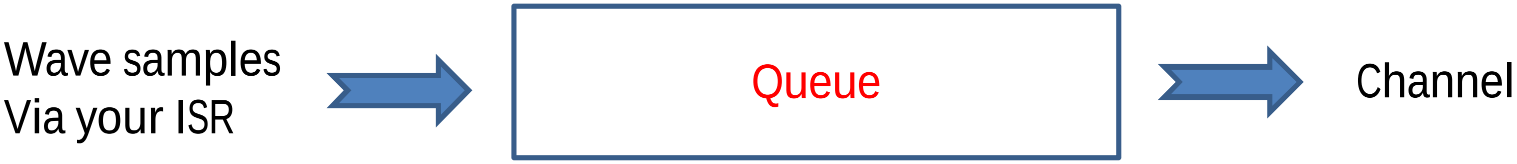


Queue

Channel

More updating

You must provide the queue with samples often enough that it never gets empty. An interrupt will happen when the queue gets close to empty, at which time you will fill it up with the next samples from the wave you are generating.



Coordination

If you are doing a square wave, a counter can tell you when to change from the positive output to the negative output and back.

Note that you continue to fill the queue from where you left off; you do not restart at the beginning of the sound! You could find you are part-way through a wave when the queue becomes full, and you need to start at that part-way point when the next queue interrupt comes.

Multiple Frequencies

For multiple frequencies of sound going down the same channel:

Add the values! Be sure the absolute value of the sum does not exceed 32,767 and go beyond the allowed range! (Limit the size of the summands!)

Note that zeros into the channel means “off”.

Example

Say you want to output a square wave with frequency of 1KHz.

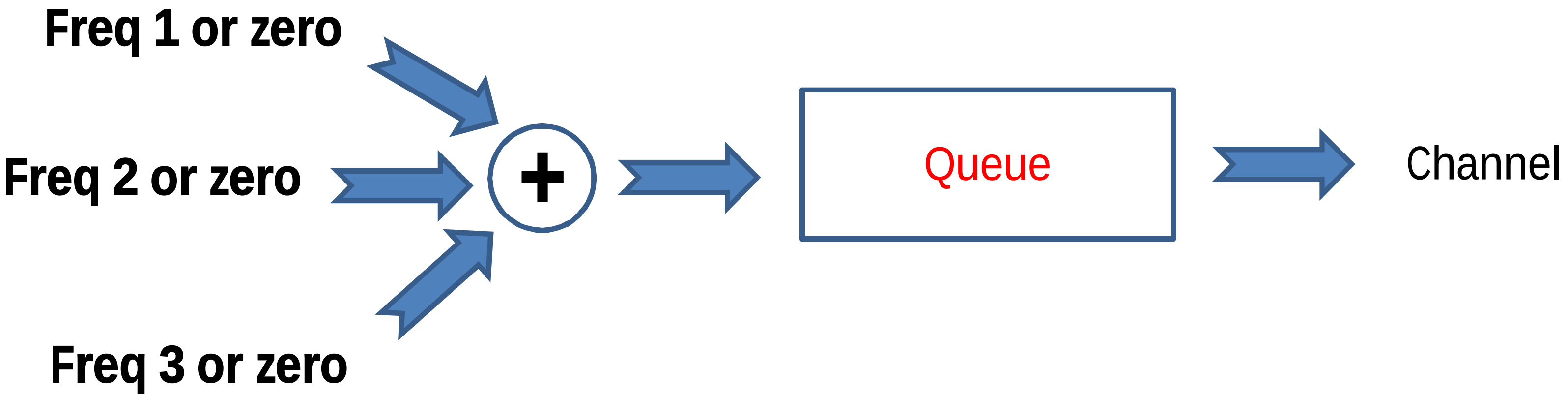
This is 1/1000 seconds per cycle, or a change from high to low or low to high every 1/2000 sec

For a 44 KHz output rate to the channel, this means a change every 44000/2000=22 outputs.

If we want to be able to add 3 different frequencies together and not overflow, we will pick a volume of 10,000 for this frequency.

So every 22 outputs we will switch from loading a value of 10000 into the channel to -10000, then switch back after loading 22 outputs of -10000.

For 3 Possible Inputs



If you want the specific frequency, add it in, otherwise add in zero.