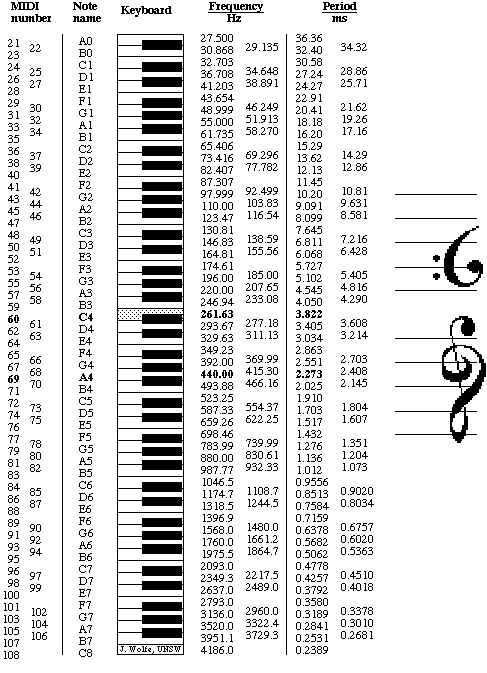


* “note\_on” tells the key is to be pressed (or released, if velocity=0).
* “note\_off” tells the key is to be released (velocity should always be set to 0).
* “channel” tells to which channel the sound is to be sent. The standard midi supports 16 channels simultaneously.
* “note” tells which key it is. We can refer to the map below for the corresponding key on piano keyboard to each midi note id.
* “velocity” tells how fast to strike the key, the faster it is, the louder the sound is.
* “time” tells us the waiting time between the last and current operation. The duration of a note is the sum of “time” from each message in between of 2 nearest messages about the same note, where the first one tells you to on the note (when you see “note\_on”, and “velocity” > 0) and the last one tells you to off the note (when you see “note\_off”, or “note\_on” with “velocity”=0).



(<https://medium.com/analytics-vidhya/convert-midi-file-to-numpy-array-in-python-7d00531890c>)

* A MIDI track is created by a software instrument
* The track contains data about notes and playback sound
* MIDI stands for Musical Instrument Digital Interface
* It’s easy to edit notes on a MIDI track
* A MIDI file ends with .MID
* MIDI file makes no sound. It simply tells the software what to play and how to play it, the same way a pianola roll tells a pianola which notes to play and how to play them
* Music file that contains information about notes: which pitch, when to play them, how to play them, velocity, length
* When you open this file with a music software program, the software knows what to play
* The software uses its own sounds to play back the file
* The file can sound different when opened in different software apps because each app has its own unique set of instruments

1. why are there duplicates?  
   (2) are the changes in tempo at different delta times important ? (ie. is this because the music speeds up/slows down in various parts  
   (3) Is it worth implementing a hidden markov chain that deals with tempo changes for my generator

For a format 0 file, the tempo will be scattered through the track and the tempo map reader should ignore the intervening events

I think this means the first tempo is the one to be used. I've dragged midi files with multiple tempo events to multiple programs, and they tend to pick out the first tempo, although Ableton seems to be kind of random about which one it chooses sometimes.

My midi file also has a long string of "set tempo" messages. They appear to be dummy messages, which are only used for their 'time' field values.

The time fields are being used to place "marker" and "key\_signature" messages at specific times in first track. This first track has no notes in it. See below. The second and third tracks have notes in this midi file (these two tracks are not shown below).

What the PPQ means in terms of absolute time depends on the designated tempo. By default, the time signature is 4/4 and the tempo is 120 beats per minute. The tempo is expressed as a 24-bit number that designates microseconds per quarter-note

Note-off with velocity: release velocity

some samplers use note-off velocity with "on release" trigger types, useful for reproducing more natural release or reverb trails. Shortcircuit is one such sampler you can freely test this on.

Only very few keyboards provide note-off velocity. Most just send Note-On with velocity 0 instead of a note-off message. So supposedly all hardware and virtual instruments accept Note-On with velocity 0, and only very few will do anything sensible with Note-Off with changing velocities.

I don't understand where the note-off velocity values you want to send to the sampler should be originated from. If course you rather easily could do a JSFX that converts each Note-On with velocity 0 to a Note--off with a random velocity. No idea how this might sound like.

this is simply because that is what a lot of MIDI keyboards send out. You can do either according to the MIDI spec. The advantage of using note on with zero velocity is that it can be sent more efficiently if you use

It seems to be dependant upon whether a keyboard is capable of generating and transmitting note velocity.

the velocity byte in a Note Off message was intended to be available for use in adding expression to the ending of a note, .e.g. perhaps modulating the release time, or filter release curve. Synths not implementing this ignore the velocity byte and just terminate the note the same way as a Note On with zero velocity. However I never came across anything that actually implemented Note Off Velocity (not that my experience is particularly encyplopedic).