Brief description of MIDI

The Musical Instrument Digital Interface (MIDI) specification is a hardware and protocol standard that was developed to provide a common communication system between musical equipment. Not to be confused with the MIDI file type, which is used to record a sequence of MIDI events like a song.

Even if you are not planning on using any external MIDI equipment, you will find that a lot of the SoundFont specification is based on MIDI.

MIDI compatible devices that wish to communicate with each other are connected by cables (2 cables, one for each way of the traffic) and if all goes well the chatter of the MIDI protocol ensues.

Most wavetable cards (and some regular sound cards) are MIDI capable.

Usually there is a special cable that hooks to the joystick port of your sound card with MIDI connectors (standard 5 DIN a la PC Keyboard connector) and a joystick connector so you can still play your games.

The protocol defines things like “note on” and “note off” events as well as controller change messages. An example of a controller is the “pitch bender” which is commonly found on keyboards and is reserved its own controller number, but the MIDI standard also allows for arbitrary controllers such as: pedals, levers and hamster wheels, you name it.

Note on/off events contain information about which note the action is taking place on as well as how fast the note was played (called velocity). Notes are specified by a number in the range of 0-127 with 60 being “middle C”. Velocity is also specified on a 0-127 scale (127 being the fastest key press). These messages are sent on individual channels, of which the MIDI specification has 16 of them. A channel is used as a way of routing MIDI messages to specific devices on your MIDI chain (as there may be several). Each device is set up to listen on a certain channel, specified by the user. Channel 10 is commonly set up as a percussion channel (drums, cymbols etc), it is the only channel that has a common assignment, the others are arbitrary.

A Preset number is used to specify which instrument to play. The original specification defined 128 presets in the range of 0-127. There are different standards for how Presets map to instruments. Under the GM and GS preset standard preset 0 is Piano, preset 4 is Electric Piano, 6 is Harpsichord, 126 is a Helicopter, etc. 128 individual presets is pretty limited though, so the idea of a Bank was created.

The bank number is in the range of 0-127, Bank 0 being the GM and GS “Capitol tones” as described above. Each bank is a set of 128 presets, which can be variations on the standard Bank 0 or be its own unique set of instrument sounds. The GS preset standard has several variations of the standard Bank 0 presets, whereas in Bank 0, Preset 124 is the sound of a Bird, Bank 1 Preset 124 is a Dog, and Bank 2 Preset 124 is a Horse-Gallop which all have to do with sounds of animals, they are variations on the original Bird sound. There is also an MT-32 preset standard.

Its important to understand that these are just standards created that define the mapping between preset numbers and instrument sounds. This mapping is arbitrary though, and as the standard GM, GS, MT-32 sounds get kind of boring quickly, its good to know that you can define an instrument sound to be whatever Bank and Preset combination you want, using SoundFont files.

Technical note: In fact you’ll find most of the parameters surrounding the MIDI standard have a range of 0-127. This is because the MIDI standard only uses 7 bits of each byte transferred along the MIDI bus (your cables). The remaining bit is set to 1 for the command (op code) portion of a message, and 0 for the command’s operands. MIDI messages that need more bits of resolution will send multiple bytes which are combined, such as the pitch bender which uses 2 bytes for a total of 14 bits of resolution.

Samples

Samples make up the heart of the SoundFont file. They are composed of digital audio, like a .WAV or other type file, though they can also be loaded from ROM on the wavetable device (the AWE 32, for example, has 512k of preloaded ROM samples). Instruments with a continuous sound, like a flute, often have periodic patterns that can be reproduced by looping a portion of the sound. Therefore samples in a SoundFont have the option of being looped, allowing for realistic reproduction of continuious sounds by recording only a portion of the original "real world" sound. So to continue the example of a flute, the sample would have the initial breath sound and then loop for the rest of the duration of the instrument.

Instruments

Instruments use one or more samples combined with effects generators to create a sound producing device. Generators, in other words "parameters", control properties of a sample such as initial pitch and volume as well as how these parameters are affected over time. The collection of a sample and the generators that affect it is called an instrument zone. There is also a special type of zone called a Global Zone which contains only generators, which sets default generator values for instrument zones that don't specifically set those generators. A global zone is optional and there can only be one of them. Heres a more visual representation of an Instrument as a tree:

```
 |
+- Overdrive Guitar          <-- Instrument
 | |
| +- Global Zone             <-- Global Zone
| | +- Reverb = 20%          <-- Generator, amount of Reverb effect
 | |
| +- Metal Gtr E1            <-- Instrument Zone w/ sample "Metal Gtr E1"
| | +- Note Range = 0-44     <-- Generator, range of notes zone is active on
 | |
| +- Metal Gtr D2            <-- Another Zone, sample is "Metal Gtr D2"
| | +- Note Range = 45-48
| | +- Reverb = 33%          <-- This zone specifically sets Reverb
```

This is a nice ascii representation of an instrument called "Overdrive Guitar". It has 3 zones, 1 is a global zone and the 2 others are samples called "Metal Gtr E1" and "Metal Gtr D2". Most quality instruments in SoundFont files are composed of multiple samples, at different tones, of the instrument it is trying to reproduce. Each of these samples occupy a portion of the note range that is closest to their original recorded tone. The reason for this is that pitch shifting a sample (raising or lowering the tone) is a lossy process, as digital samples are of finite resolution (digital sampling is the process of taking individual "slices" of a sound at regular intervals).

Too much increase or decrease in tone will cause a sample to sound less like the instrument it is trying to reproduce. In this example the higher pitched sample "Metal Gtr D2" takes over from "Metal Gtr E1" at note 45. Each of these zones has its own generators. The "Global Zone" sets the value of the Reverb generator to 20%, globally. So any instrument zone that doesn't specifically set the reverb generator will be assigned a reverb value of 20%. In this example the "Metal Gtr E1" zone does not set the Reverb generator so it will be played with 20% of the reverb effect, whereas the “Metal Gtr D2” zone assigns reverb 33% so the “Global Zone” does not affect it.

Presets

“So if you have an instrument already, whats the use of a preset?” you ask. Well, once you have your instruments you need to assign them to MIDI bank and preset numbers, and sometimes its nice to make other variations of your instruments without having to duplicate them. This is where presets are used. A preset is like an instrument in that it has zones and generators, but instead of having a sample in each zone, it has instruments (except for a global zone which just has generators). The generators also act differently when they are in a preset zone. While instrument generators set a parameter to the absolute value you specify, preset generators offset the values of every instrument zone in the instrument that is part of the same preset zone. Here is another ascii image, this time of a preset that uses the “Overdrive Guitar” instrument:

|
+- 000-029 Overdrive Guitar <-- Preset on Bank 0, Preset 29

| |
| +- Overdrive Guitar <-- Preset Zone w/ instrument “Overdrive Guitar”

| | +- Note Range = 10-127 <-- Generator, Note Range of this Zone | | +- Reverb = 10% <-- Generator, amount of Reverb effect

| | +- Chorus = 15% <-- Generator, amount of Chorus effect

In this example the Instrument created in the above example is assigned to Bank 0, Preset 29. 3 generator parameters are modified; Note Range, Reverb and Chorus. The values of these generators act as offset values which are added to the values of the instrument “Overdrive Guitar”. So in the Instrument example we determined that the Reverb effect for each zone was:

Metal Gtr E1: Reverb = 20% (set by Global instrument zone) Metal Gtr D2: Reverb = 33%

Since the value of the Reverb generator in the preset is 10%, this amount will be added to each instrument zone of “Overdrive Guitar”. So the values will then become:

Metal Gtr E1: Reverb = 20% + 10% = 30%

Metal Gtr D2: Reverb = 33% + 10% = 43%

Note that if the result of an addition goes out of the valid range of a generator (0-100% for Reverb) then it is clamped to its valid range.

In the case of the Chorus generator in the Preset zone, the Chorus parameter was not set in any zones of the instrument. Each generator has a default value that is used if no values are specified. In the case of Reverb and Chorus they default to 0%, or no effect. So the chorus for each instrument zone then becomes:

Metal Gtr E1: Chorus = 0% + 15% = 15%

Metal Gtr D2: Chorus = 0% + 15% = 15%

The affect of a Note Range or Velocity Range in a preset zone is not of an offset or additive nature. Instead the ranges defined in an instrument zone are intersected with the range of the preset zone. So only notes for which both the Instrument Note Range and Preset Note Range are defined become active notes after the intersection. The note ranges in the Instrument example were:

Metal Gtr E1: Note Range = 0-44

Metal Gtr D2: Note Range = 45-48

The Note Range is 10-127 in the Preset zone. So the intersection of the instrument zone ranges with the preset zone range becomes:

Metal Gtr E1: Note Range = 0-44 intersect 10-127 = 10-44 Metal Gtr D2: Note Range = 45-48 intersect 10-127 = 45-48

More on generators

Each generator has a specified parameter format, set of allowable values, and a default value. These values are different between the instrument and preset levels.

Example: The Reverb generator at the instrument level; is specified in percent, has an allowable range of 0-100, and a default value of 0.

At the preset level Reverb is specified in percent, has an allowable range of -100 – 100, and a default value of 0.

The default value is used if the parameter is not specified for a given zone.

The descriptions of generators below are for the Instrument level, and therefore are the absolute limits of the corresponding generator. The limits at the preset level can be determined from this information. The default value for preset generators is such that there is no affect on the instruments. For ranges this is the range 0-127, for all other generators it is 0. The allowed value of a preset generator is such that it can offset a generator at the instrument level to any portion of the generators absolute range. So for Reverb this is -100 for the low range to bring a generator to 0 if its 100, and 100 for its upper range to bring a generator to 100 that is 0. The units at the preset level are the same as the instrument level for the SoundFont standard, note however that some of the units used in Swami, and in this document, have been converted to more convenient units. This causes some generators to have a multiplicative affect rather than an additive affect. Generators with these properties have the units column set to “X” in Swami.

The following different parameter units are used:

Note – SoundFont files use the MIDI note scale, which is 128 notes numbered 0-127, and note 60 is “middle C”. An octave is normally devided into 12 steps, called notes. An octave can be arbitrarily devided though. Some other note scales do things differently, and SoundFont files give you some flexibility in this area. An octave is basically a two fold increase in frequency. So normally note 60 “middle C” is twice the frequency of note 48 which is the next lowest C note.

This is why playing two of the same notes “resonate” with each other, its because the waveforms are in sync with one another. When one note completes a single cycle the other completes 2, 4, 8 etc (also known as harmonic frequencies).

Percent – The percentage the affect of a generator has.

Semitones – A semitone defines a change in pitch and is the difference between two consecutive notes (counting flats and sharps). So a pitch difference between note 60 and 61 (Middle C and C#) would be 1 semitone. This is abbreviated in Swami as “semi”.

Cents – A cent is $1/100^{th}$ of a semitone and is used for defining smaller changes in pitch.

Decibels – A decibel is a logarithmic unit of amplitude measurement.

Logarithmic units are used commonly in audio because human hearing tends to react on a logarithmic scale, rather than a linear scale. In other words a sound that is twice as high in amplitude won't sound twice as loud, or any other determinable linear fraction. I often get confused by the decibel, if you have this trouble too, just try to remember that it is just a scale that more closely models things like hearing. In mathimatical terms its the 20^{th} root of 10. Abbreviated as "dB" in Swami.

Hertz – Named after some dude I believe. Think "something" per second, I.e. number of times something occurs per second. Frequency, or tone when you hear it, is often specified in Hertz (abbreviated "Hz"). The average lowest frequency the human ear can perceive is 20 Hz, basically a sine wave that goes through 20 periods in a second. Digital sampling rate is also often specified in Hz. CD quality is a sampling rate of about 44100 Hz, so the audio signal amplitude is digitally "sampled" 44,100 times per second.

Seconds – Ummm.. I hope you know this one. Just in case you don't use seconds, happen to be from another universe (like me), or are lucky enough to live in the Burmuda Triangle with an appropriately translated copy of this HOWTO.. A unit of time corresponding to approximately $1/86400^{th}$ of the time it takes for the Earth to rotate once.

Abbreviated "sec" in Swami.

Misc. Generators

The Root Key generator defines which note will play the sample at its original recorded rate (I.e. sampling rate).

Example: A sample is recorded at 44.1khz (44100 samples per second, CD quality) and the root key is 60 which is middle C. Playing note 60 would cause the sample to be played at its original sampling rate of 44.1khz.

Playing a note higher than the sample's root key would increase the playback rate causing the sample to be of higher pitch, and a lower note would cause, surprise, a lower pitch.

The Note Range generator defines the note range a zone is active on, specified as a low and high note number.

Example: 60-80 would be note 60 through 80 inclusive.

Only notes in this range will cause the zone's corresponding sample to be heard. Samples in an instrument are layered when they occupy the same note(s). So if a note is played and two or more instrument zones are active for said note then all are played simultainiously.

The Velocity Range generator defines the range of velocity levels the sample is heard like Note Ranges. Velocity is the speed at which a note is played, this applies mostly to velocity sensitive MIDI keyboards.

Playing a note "softly" is a low velocity level while "banging" on the keyboard is a high velocity level.

The pitch can be modified by 3 instrument generators. Course Tune changes the tone of the sample by semitones. Fine Tune modifies the tone by cents. Scale Tune defines how the tone changes from note to note in cents. This is the parameter that allows you to create other note scales besides the default 12 note octave devision. Default is 100 which is 1 semitone.

Effects Generators

Filter Q and Filter Cutoffcontrol the initial state of the filter effect. The filter in this case is a low pass filter (passes lower sounds and blocks higher ones), commonly used/abused in electronic music. Also known as the “wah-wah” effect.

Filter Cutoff is the frequency at which frequencies above it start to be significantly blocked by the filter. The default is aproximately 20khz which would pass all sounds below this frequencly relatively un-molested, since 20khz is about the highest tone an average human can hear, the filter would basically be off. Lowering this value to say 2khz would start to significantly block all frequencies in the sample above 2khz.

Filter Q affects the gain at the resonant frequency of the filter. I don’t think this has much to do with Q in the sense of Quality of a filter. It looks like increasing this value causes a sudden increase in gain at the resonant frequency (aprox. The Cutoff Frequency). So with a high Filter Q: sound below the Cutoff frequency would be relatively un-affected, sounds at and around the Cutoff frequency would be accentuated (louder), and frequencies above this would drop off rapidly (blocked). Basically brings a small range of frequencies around the Cutoff into the foreground.

Reverb adds a short delay “echo” effect to the sample. Increasing this value will make the sample sound like its in a hall or cave.

Chorus gives a sample a “fuller” sound. Like the term suggests it makes the sample sound like several duplicates of the sample being played at the same time but at slightly different tones.

Pan sets the left/right balance of the sound. Good for making stereo instruments, where one sample might be placed more to the left and another slightly different sample, to the right.

Envelopes

The idea of an envelope is not unique to SoundFont files. They are used quite frequently in sound producing devices. An envelope defines how a parameter changes through the various stages of playing a note. The envelopes used in SoundFont files have 6 stages listed below:

Delay – Amount of Time before the attack stage begins, the envelope stays at its zero level

Attack – Time it takes for the envelope to reach its max value from when the key is pressed

Hold – Time the envelope is held at the max value
Decay – Time it takes for the envelope to decrease to the Sustain value
Sustain – Value the envelope reaches at the end of the Decay period
Release – Time it takes for the envelope value to decrease to zero after the key is released

As an example, lets say an envelope is controlling the volume of a sound, though the parameter (volume in this case) is arbitrary. To begin the envelope cycle, your pet ferret steps on a single key on your MIDI keyboard. Nothing is heard during the Delay stage for the period of time set by the Delay time. Next, the volume of the sound begins to rise until it eventually reaches its highest attainable loudness (to your ferret’s delight who prefers loud sounds). This is the attack stage, and the Attack time controls how long it takes for the sound to reach its maximum volume. Now that it has reached its max volume it stays at this level for a period of time determined by the Hold time.

Apon the end of the Hold stage the volume level begins to decrease for the Decay stage. The value it is going to decrease to is determined by the Sustain value, and the amount of time it will take to decay to this value is determined by the Decay time. At this point the volume is in the Sustain stage and will remain at the Sustain value until your ferret gets bored with that tone (or you throw something at it) and releases the key. This starts the Release stage where the volume will decrease until the note is off, in the amount of time set by the Release time. Look at the illustration of an envelope to get a visual idea.

Volume Envelope

The volume envelope is basically what was used in the example above.

This envelope controls only the volume parameter of a sound. There are 6 parameters for each stage of the envelope. Delay, Attack, Hold, Decay, Sustain and Release. Every parameter is specified in seconds except for Sustain which is in Decibels (dB). Sustain is the amount of volume attenuation (decrease) from the maximum volume, during the Sustain stage. Setting this to a higher value results in lower volume during Sustain, zero is no attenuation (maximum volume). There are 3 other parameters affecting volume; Attenuation, Key to Hold and Key to Decay. Attenuation sets the initial volume attenuation of the sound before being modified by the envelope (basically controls what the max volume can be, the envelope can only make it quieter). Key to Hold and Key to Decay will be explained later (when they actually work in Swami :), but basically they control the degree of decrease per MIDI note for the Hold and Decay stages respectively, I have yet to have used them for anything.

Modulation Envelope

The modulation envelope controls two parameters; Pitch and Filter Cutoff. The 6 envelope parameters are the same as the Volume envelope except for Sustain which is now in percent (rather than dB). Sustain in this case is the amount of the effects (Pitch and Filter Cutoff) that are applied to the sound during the Sustain stage. The “To Pitch” parameter controls the maximum amount of pitch change (when the envelope is at the end of the Attack stage) and it is specified in cents. When it is at a zero level the pitch is not affected by the modulation envelope. If “To Pitch” is positive then the modulation envelope will increase the pitch by a maximum of “To Pitch” cents.

Conversely if its negative then the modulation envelope will decrease the pitch by the maximum specified cents.

Low Frequency Osciallators

A low frequency oscillator (LFO) is just a sine wave generator that has a relatively low range of frequencies, compared to human hearing, that it operates at. In a SoundFont there are two LFOs, one of them controls just the pitch of the sample while the other also controds the pitch as well as the volume and filter cutoff parameters.

Vibrato LFO

Vibrato is a term used in music, its a rapid oscillation of the pitch of a sound. An opera singer uses quite a lot of vibrato. The vibrato LFO controls only the pitch of a sample. It has 3 parameters; Delay, which is the amount of time before the oscillator starts (in seconds); Frequency, which is the frequency of the vibrato; and “To Pitch” which is the amount of vibrato, I.e. the amplitude of the oscillator (in cents).

Example: Setting “To Pitch” to 0 will essentially disable this LFO and no vibrato will result. Setting it to 400 cents would cause a note’s pitch to oscillate between 200 cents (2 semitones) above the note to 2 semitones below the note. Setting “To Pitch” to a negative value will basically shift the phase of the effect by 180 degrees. When the LFO goes up, the pitch would go down, and when the LFO goes down, the pitch would go up.

Modulation LFO

The modulation LFO can control the pitch, volume and/or filter cutoff of the sample. I'm not sure the exact reasons for letting both LFOs control the pitch, but it does seem to me to be the most worthy of effects. Some rather psycho alien sounds can be made from just about any sample by setting both LFOs to tweak the pitch, but be careful you don't activate your alien implant. The Delay, Frequency and "To Pitch" parameters are just like the Vibrato LFO. The "To Filter Cutoff" parameter is the amount of affect the LFO has on the Filter Cutoff parameter (in cents). The "To Volume" parameter is the amount of affect the LFO has on the volume (in dB). If you can't imagine what these effects sound like, I'll describe them. If the "To Volume" parameter is set to something besides 0 (disabled), then the volume of the sample will pulsate up and down at the rate defined by the Frequency parameter, I think this is called "portamento" in the music world. If the "To Filter Cutoff" parameter is set to something besides 0, then the cutoff frequency of the low pass filter (described in a section above) will pulsate up and down at Frequency, causing higher frequencies to be blocked as the cutoff parameter goes down, and allowing more higher sounds when the cutoff goes up. Sometimes called a "Wah-Wah" effect. Must hear it to know what they mean.

Modulators

Modulators are used to change the value of generators in real time, in response to various control values such as velocity and custom MIDI controllers. This could, for example, allow you to make the "pitch bender" controller on a keyboard modify the Reverb generator instead of the pitch. Swami and iiwusynth have support for modulators now, so I will finish this section of the document soon :)