SoundFont 2.1 Application Note

Joint E-mu/Creative Technology Center
E-mu Systems, Inc.
Wednesday, August 12, 1998

Abstract

This document explains SoundFont 2.1 and how to take advantage of the features that it provides.

The MIDI Standard

The MIDI protocol gave us a wonderful mechanism from which audio could be generated and controlled. Since the MIDI protocol only sends information pertinent to run-time control over a sound, rather than the continual updating of the complete definition of the sound, it allows for a realistic music experience and at the same time serves as a form of audio compression.

Unfortunately, MIDI standards were not strict enough about the definition of what a sound is. General MIDI only defines the sound set which should be used with the MIDI protocol. For example, General MIDI says that Program 1 should be a piano. What General MIDI does NOT do is describe how that piano should sound. Also, although General MIDI has been extended by Roland (with GS) and Yamaha (with XG) to add more sound to the basic palette, General MIDI can never be extended in this manner to accommodate all possible sounds.

Another problem with MIDI is in the area of real-time control of sounds. The MIDI standard is extremely strict about the scope of how a sound may be controlled in real-time, and at the same time the same MIDI standard is vague about the precision of how those parameters are controlled in real-time. Real-time controllers may only take effect globally on a program (equivalent to an entire instrument) and could only modulate basic parameters, including pitch, volume, and effects. At the same time, the precise manner by which these parameters are controlled (such as the resolutions, ranges and curves) is vaguely defined in the MIDI standard. Manufactures of synthesizers which are designed to be MIDI compliant are forced to reverse-engineer the resolutions, ranges and curves of one of the widely adopted MIDI synthesizers in order to become "compatible."

Thus the problems which resulted… MIDI sounds different on different synthesizers because there is no way to define the way an instrument sounds, and the MIDI protocol limits the composers ability to enhance the sounds in real-time. These are the primary reasons why MIDI compositions typically sound cheesy and unrealistic.

The SoundFont 2.0 Standard

SoundFont 2.0 has already solved the first half of this problem.

It provides a way with which one could define a sound, and a way with which such a sound could be sent to a synthesizer. Once the sound is sent to the synthesizer, that sound could be controlled with the MIDI protocol.

Since wave-table synthesis was, still is, and will likely continue to be the most practical way to define high quality electronic musical instruments, SoundFont 2.0 defines a sound as a collection of samples grouped into instruments which could be further grouped into presets. Instruments and presets could have individual volume, pitch, and filtering settings as well as defined envelopes and low frequency oscillators to further modulate those parameters.

So now, we have a way to say “this is what a MIDI controlled piano will sound like” using a definition which may be supported by any synthesizer with a typical wavetable synthesizer architecture model. This was a step forward, but it only solved half of the problem.

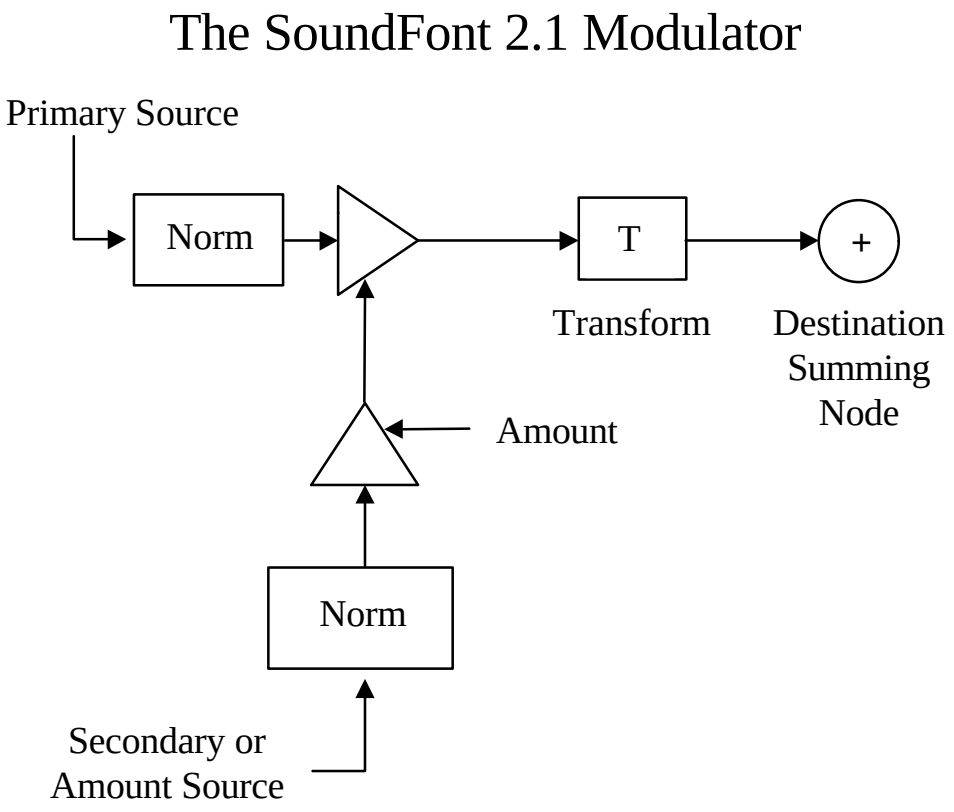
MIDI still limits the ability to control your sound. Controllers are either ambiguous in nature, or very strict about how they influence the synthesized audio. There is no definition on how Modulation Wheel or Breath Controller influences a given sound. For different MIDI programs, these controllers should really have different effects that complement those programs. Pitch wheel, on the other hand, MUST influence pitch for ALL sounds contained within a single program on a single MIDI channel. So if a composer wants to use sound comprised of two layered samples and wants the same pitch wheel to bend the first sample up and the second sample down, he cannot do it. He is often told by strict MIDI advocates to use two different channels. Unfortunately with only 16 MIDI channels, tricks like this may not be used very much. And… that is a tremendous waste of development resources.

The SoundFont 2.1 Standard

SoundFont 2.1 will solve the second half of this problem by introducing the *modulator* to the SoundFont format. A modulator is a definition of how a particular synthesis parameter will be altered in real-time as a result of a user or system input.

In other words, SoundFont 2.1 will allow one to say “this is how a SoundFont 2.0 preset may be altered in real-time, and how it will sound like when it is altered in real-time.”

To see how this is done, we will start with the basic structure of the modulator:



Primary Source:

The source is what causes an effect to occur on a synthesis parameter.

A source may be any one of a wide range of MIDI controllers and other system driven events. A source may also be a synthesizer driven source (such as an envelope or an LFO) in the event that those outputs were made available to the SoundFont rendering engine.

The Sources currently supported by SoundFont 2.1 are the following:

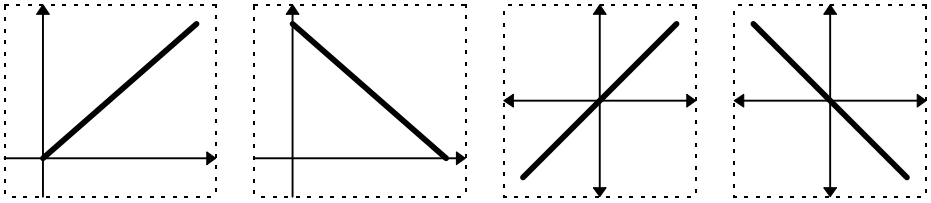
Key On Velocity
Key Number
Pitch Wheel
Channel Pressure
Pitch Wheel Sensitivity
Modulation Wheel
Breath Control
MIDI CC 7
MIDI CC 10
MIDI CC 11
MIDI CC 21
MIDI CC 22
MIDI CC 23
MIDI CC 24
MIDI CC 91
MIDI CC 93

Sources have a well-defined minimum and maximum values, and have linear resolutions. They go through a “normalization” stage that does two things: first, converts the source native units to arbitrary values between -1 and 1, and second, re-maps them to a user selected characteristic curve.

The mappings currently supported by SoundFont 2.1 are the following:

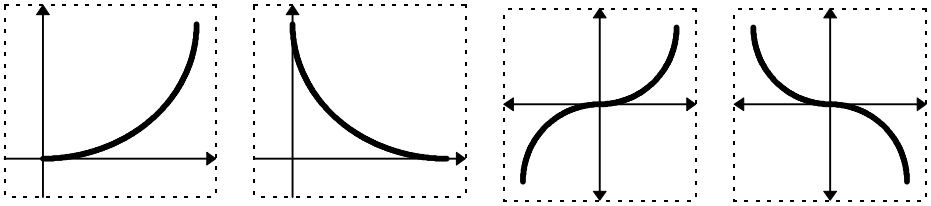
Source Controller Type Summary

Positive Unipolar | Negative Unipolar | Positive Bipolar | Negative Bipolar



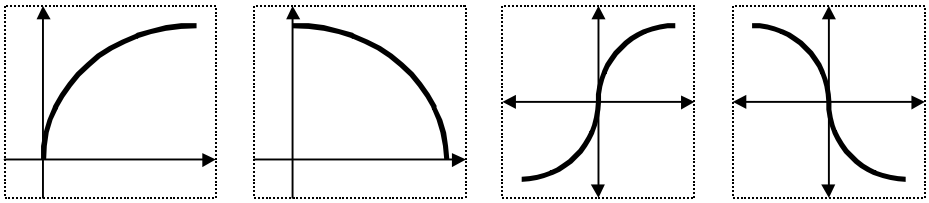
Linear Controller Curves
for given Directions and Polairities

Positive Unipolar | Negative Unipolar | Positive Bipolar | Negative Bipolar



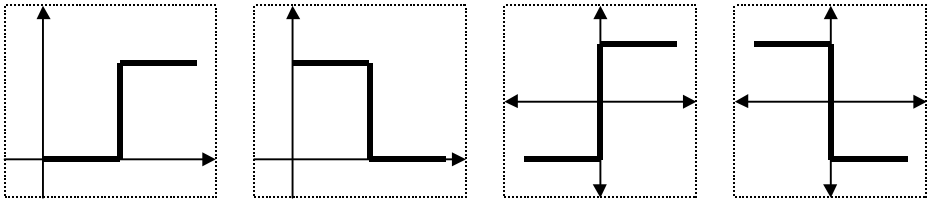
Concave Controller Curves
for given Directions and Polairities

Positive Unipolar | Negative Unipolar | Positive Bipolar | Negative Bipolar



Convex Controller Curves
for given Directions and Polairities

Positive Unipolar | Negative Unipolar | Positive Bipolar | Negative Bipolar



Switch Controller Curves
for given Directions and Polairities

The negative unipolar ‘concave’ controller curve obeys the “amplitude squared” characteristic used by the widely adopted MIDI synthesizers as a velocity curve influencing volume. SoundFont 2.1 adds a

mathematical formula to the specification that may be used to precisely compute the characteristic of the negative unipolar concave curve. The other concave and convex curves are simple variations of that characteristic.

The ‘switch’ controller curves transition at the middle point of the controller, there is no slew between the minimum and the maximum.

Secondary (Amount) Source:

A second source may simultaneously modulate the given destination. Features and implementation are identical to those in the source.

Destination Summing Node

A destination is the synthesis parameter that is being influenced by the source.

The range of destinations spans the range of generators that may be used at a given level of the SoundFont hierarchy. In other words, any of the parameters that were available in SoundFont 2.0 may be modulated in SoundFont 2.1.

Transforms

A secondary mapping for the controllers. Transforms are typically mathematical formulas that know of no notion of “minimum” or “maximum” values.

Example: MIDI Pitch Bend

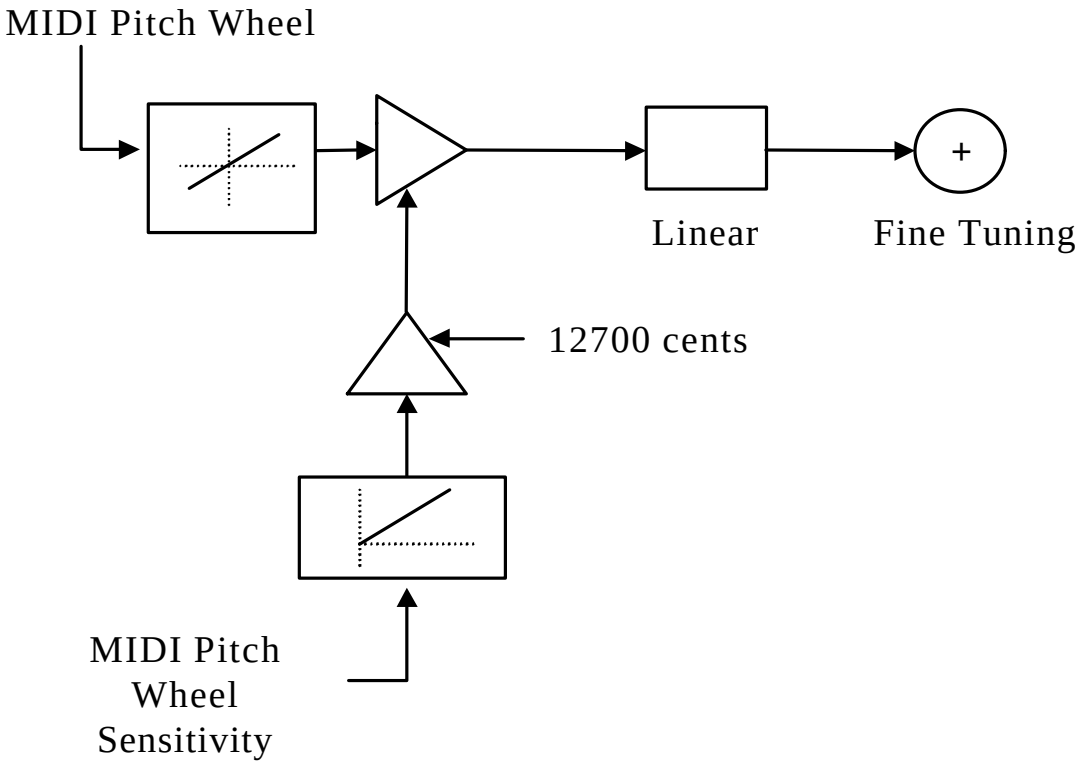
Let’s move back to our MIDI pitch bend example by first demonstrating how the modulator may be used to create the MIDI definition of pitch bend.

MIDI Pitch Bend is taken for granted that it will influence the pitch of the sound in a linear fashion with 1 cent resolution.

In reality, MIDI Pitch Bend is just a wheel which sends data to a synthesizer and that triggers the synthesizer to move pitch up and down. The only reason pitch is moving up and down is because this is what the synthesizer is programmed to do when it receives a MIDI pitch bend.

That stimulus/response can be represented using the SoundFont Controller Model in this fashion:

Example of MIDI Pitch Bend with a Modulator



Note in this picture that MIDI pitch wheel used as a positive bipolar source, meaning that the middle value (64) has an influence of **0**, the maximum value (127) has an influence of **1**, and the minimum has an influence of **-1**.

The source is routed to pitch, at an amount of **12700** cents per full swing. The reason this value is chosen is explained later.

The controller further influences pitch with another source, MIDI Pitch Wheel Sensitivity. In this case, the source is used as a positive unipolar source. This means that the minimum value (0) has an influence of **0**, and the maximum value (127) has an influence of **1**. MIDI pitch wheel sensitivity by default is set to 2 semitones or 200 cents per maximum pitch bend up. This musical based value is represented by a MIDI value of 2, or **2/127** the full swing of the possible values which MIDI pitch wheel sensitivity may move.

It is passed through a linear transform, which has no effect on the signal, and is added to the current fine tuning of the sound.

So when the Pitch Wheel is moved from the middle value (64) to the maximum value (127), it's influence is to add (**1** * **12700** * (**2/127**)) = 200 semitones to the current pitch.

Likewise, when the Pitch Wheel is moved from the middle value (64) to the minimum value (0), it's influence on pitch is to add (**-1** * **12700** * (**2/127**)) = -200 semitones.

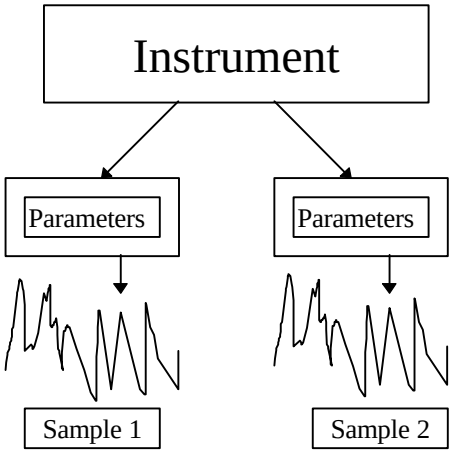
Also note that as the RPN 0 (MIDI Pitch Bend Sensitivity) controller is moved from 2 semitones (**2/127**) to 12 semitones (**12/127**), the maximum pitch bend up produces a value of (**1** * **12700** * (**12/127**)) = 1200 semitones or one octave.

Using Modulators to Break the MIDI Barrier

The above behavior is compatible with the General MIDI specification regarding Pitch Wheel and Pitch Wheel Sensitivity. However if the sound designer wants to add expressiveness to their sound in the many ways it may be outside of the constraints of General MIDI, he may do so. All he needs to do is to change the amount, or perhaps the source-mapping curve, or even the source or destination!

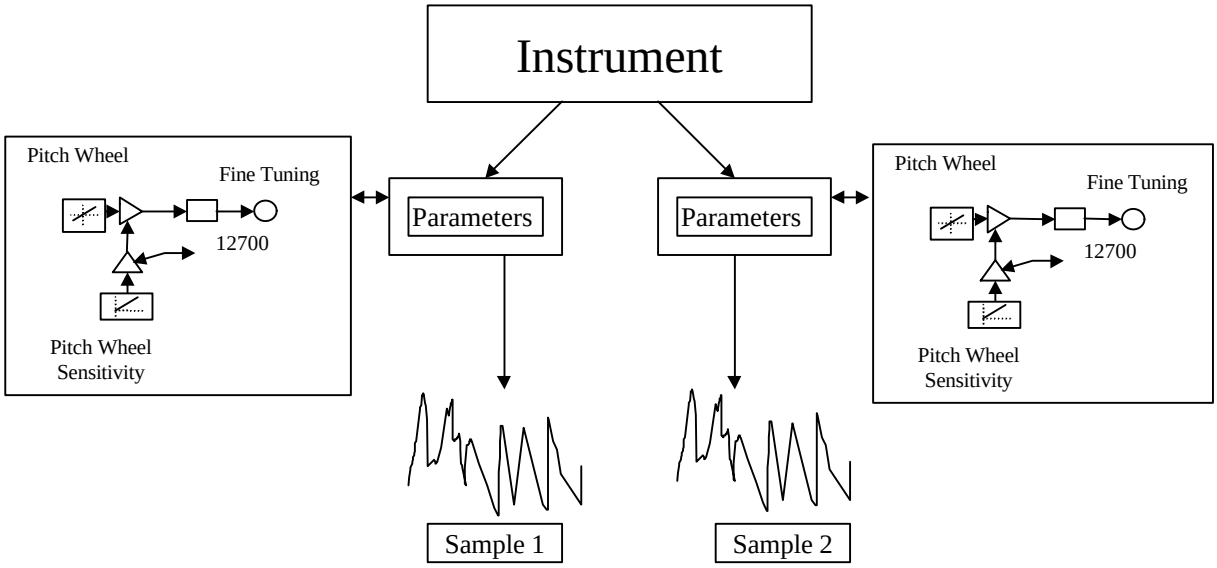
Let's assume we start with an instrument that is made up of two layered samples. Let's further assume that we have a SoundFont 2.0 instrument so there can be different parameters on each sample.

SoundFont 2.0 Instrument



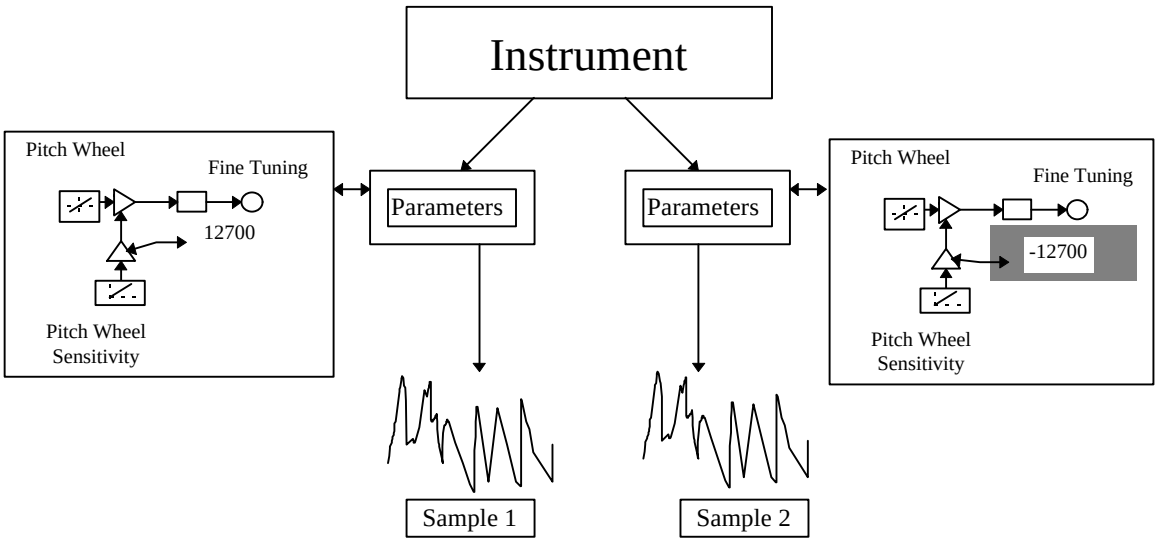
This instrument lacks modulators, so it is taken for granted that it responds to MIDI inputs such as pitch wheel in the manner that General MIDI defines. Adding the modulator we formulated above to the parameter list of each of the two samples does not alter the sound or it's response to real-time pitch bending:

SoundFont 2.1 Instrument with Explicit
MIDI compatible Pitch Bend Modulators



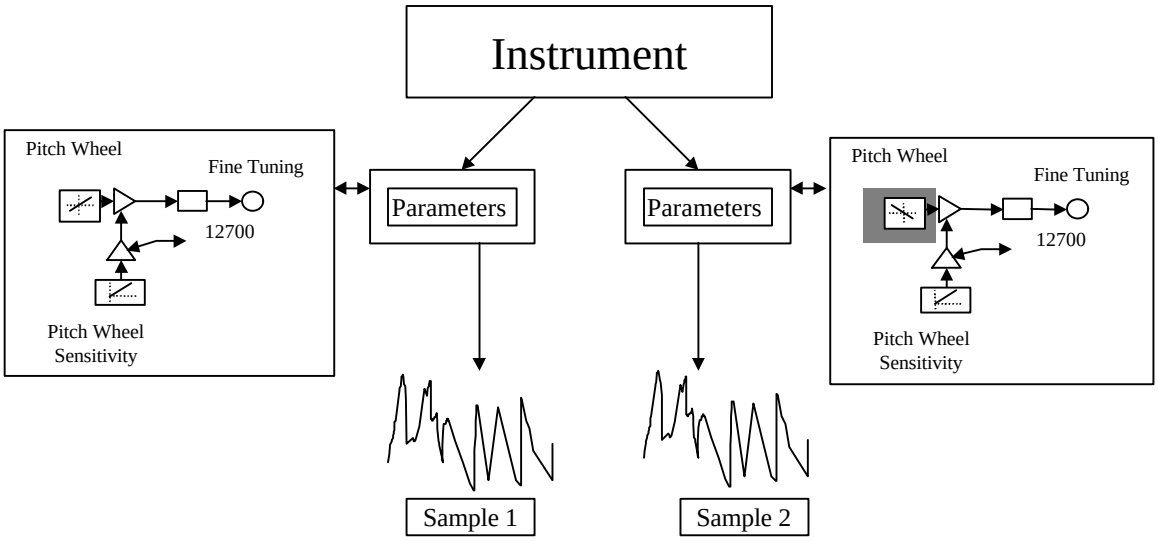
So, now that these modulators are explicitly added to the sound definition, they may be altered to produce a new desired effect, such as having pitch bend moving one sound up and one sound down. The modification… change the amount from 12700 to -12700:

SoundFont 2.1 Instrument with Custom
Pitch Bend Modulators



Or, since this is such a simple change, one could also change the source type from positive bipolar to negative unipolar:

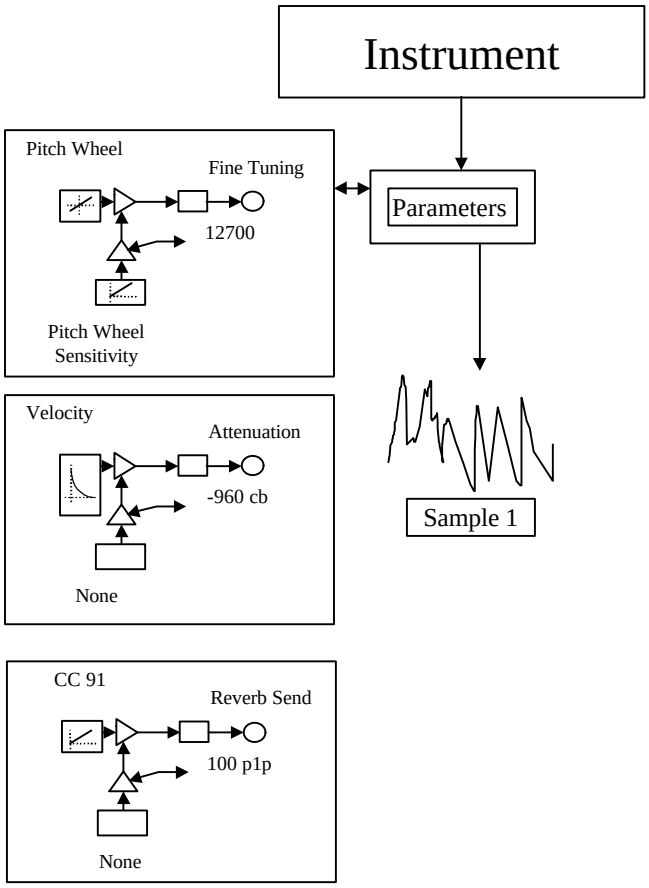
SoundFont 2.1 Instrument with Custom
Pitch Bend Modulators



Versatility

SoundFont 2.1 allows you to add as many of these modulators per instrument as you want. This allows you to create instruments which are either 100% MIDI compatible or partially MIDI compatible, or not MIDI compatible at all.

SoundFont 2.1 Instrument with Explicit
MIDI compatible Modulators



Not only that, you may route multiple sources to the same destination. If you need velocity, CC7 and CC11 to influence volume, they may do so. Also they do so in an additive manner, so one modulator won't override another. And since SoundFont 2.0 parameters are perceptually additive, the modulator effects upon the destination are mixed in the way that makes the most musical sense.

Also, you may have the same source influencing the same destination multiple ways. For example, if you have pitch bend influencing pitch two times where one is adjustable with pitch wheel sensitivity and the other is adjustable with another controller, SoundFont 2.1 allows you to do so!

This was not supposed to imply that one could not have the same source influencing the same destination in the same way multiple times… if you wanted that, you only need add the two amounts together!

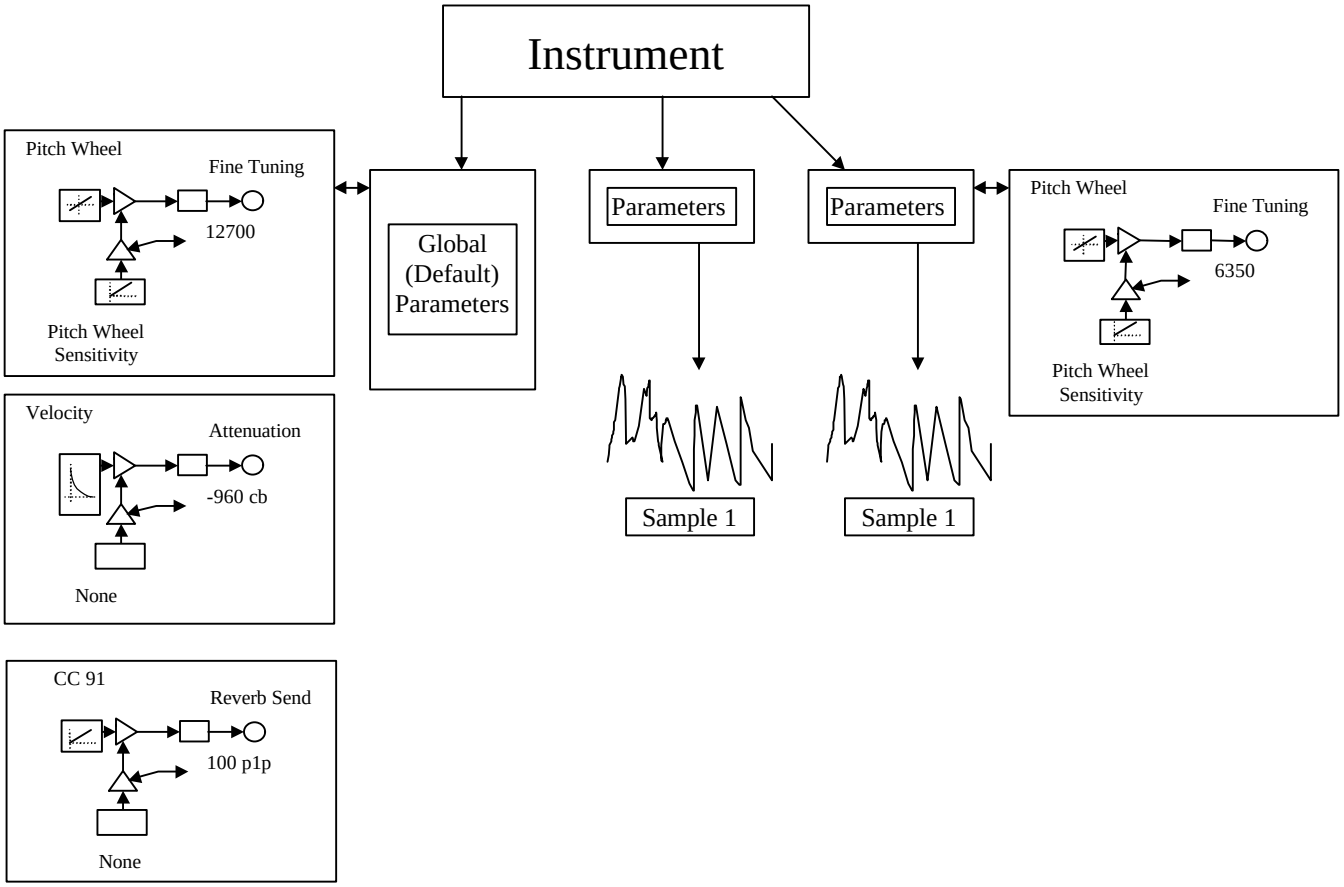
Use in all levels of editing

SoundFont 2.0 used many levels of editing to facilitate the sound designer. SoundFont 2.1 allows you to add modulators at all of these levels of editing.

Examples of using modulators in an instrument zone, in which key and velocity ranges may be assigned and synthesis parameters may added to influence samples. Examples of this were shown above.

Modulators may also be used in the global instrument zone, in which instrument default parameters may be assigned. Also Modulators in a local instrument zone override the modulators in the global instrument zone.

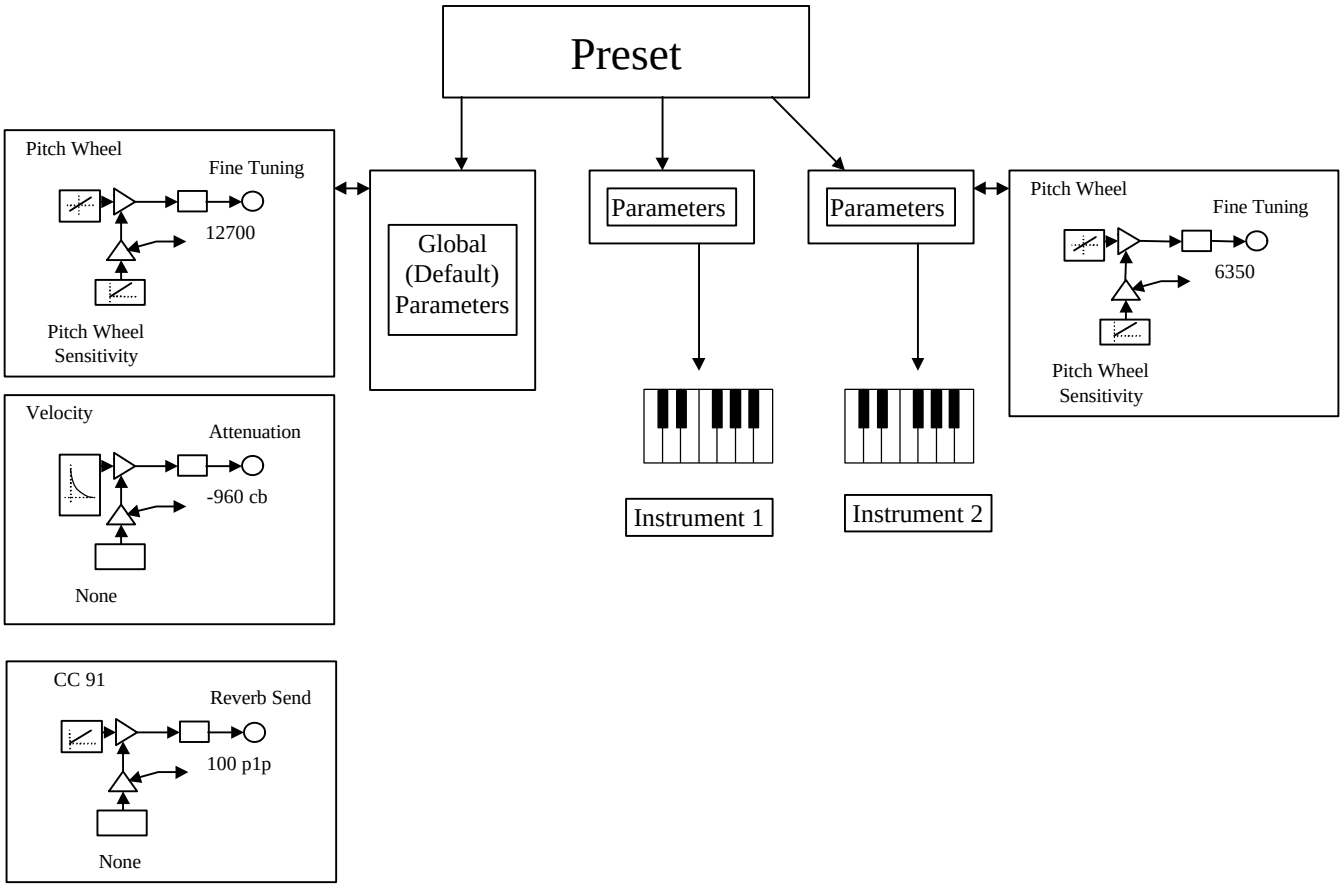
SoundFont 2.1 Instrument with
Modulators added to Global Zone and
Local Zone



The preset zone, in which key and velocity ranges may be assigned and synthesis parameters may be added to influence instruments as shown above, and the global preset zone, in which preset default parameters may be assigned.

The effect of modulators at the preset level is as follows. Preset level modulators add to the same modulators at the instrument level, if they exist. The reason to do this is much the same as the reason to use preset level generators… to enhance instruments in such a way that the instrument itself is not altered. This way, that instrument may be reused either as is or in a different manner elsewhere.

SoundFont 2.1 Preset with Modulators
added to Global Zone and Local Zone



SoundFont 2.1 NRPN implementation

In order to influence sound parameters in the many ways you can with SoundFont 2.1, the decisions need to be made at sound design time. In many situations, this makes sense, only the sound designer knows what the sound is really made up of, how many samples are used in an instrument or a preset, how those samples would sound if given certain inputs and real-time controllers. In other words, only the sound designer is in a position to make decisions as to how to influence sounds when you go beneath the single program (or preset) level.

However… there are times when a composer does not want to go through all of that just to add some real-time control over the many synthesis parameters provided by SoundFont 2.0. They simply want to add controller messages to their sequences to control these parameters.

Unfortunately, the MIDI standard precludes such composers from using standard continuous controllers to do this, because those parameters are not considered “standard.” Examples of non-standard synthesis parameters would be envelope hold segment and LFO frequency.

To address these concerns, SoundFont 2.1 also adds a standard NRPN implementation that guarantees that at least one 14 bit MIDI controller is available to provide real-time control for each and every SoundFont 2.x synthesis parameter. This controller influences the entire preset globally, which only

makes sense because the MIDI protocol does not and can not understand the details behind the definition of a single MIDI program (SoundFont preset). Any synthesizer claiming to be SoundFont 2.1 compatible will support the same NRPN implementation.

Extensible

SoundFont 2.1 addresses many issues, but not all of them. There are many more areas in which real-time controllers may be applied and used, and which are not defined by SoundFont 2.1, either because there is no hardware or software support for those features, or because it is simply impractical to do so at this time.

Fortunately, SoundFont 2.1 does not preclude such features from being added in the future. Just as SoundFont 2.1 is a purely bi-directional compatible upgrade from SoundFont 2.1, such extensions can be added in a bi-directional compatible manner.

So there is plenty of room for new features and upgrades.

Compared to the Pros

SoundFont 2.1 is not a new technology. E-mu professional samplers tone modules has been using the concept of Modulators for many years. These are called “Patch Cords” in those products, and the versatility of those patch cords is what allowed those products to play sounds with more expression and feeling than many other products in their class.

SoundFont 2.1 not only gives you much of the functionality of many of the Emulator “patch cords” with modulators, it does not add restrictions as to how many modulators may be used at any given level of the preset. Such restrictions were required for the E-mu products because they had well known and fixed CPU resources (such as local RAM and clock speed). SoundFont banks, being used mostly in multimedia applications, are used in an environment that is gaining more CPU resources by the month. So it only made sense to not add such restrictions to SoundFont 2.1, and allow the sound designer to choose which CPU speed is the minimum required in playing a particular sound.

SoundFont 2.1 does not provide all of the features that the E-mu “patch cords” feature provides. Examples of where E-mu products are superior include the usage of LFO and Envelope outputs as sources, synthesizer functions (such as cross-fade and sustain) as destinations, and the ability to cascade patch-cords. But, again, these limitations are simply a matter of definition and a matter of what may be supported with current hardware and software synthesizer technology. When the day comes that such sources need to be added to the SoundFont 2.1 model in order to account for new synthesizer technology, we’ll add them.

Availability

The E-mu Audio Production Studio (http://www.emu.com/ecard.html) will be the first product available that supports SoundFont 2.1. More will follow. Creative Technology has expressed a strong interest in incorporating the features provided by SoundFont 2.1 into the SoundFont compatible Sound Blaster series of products, however a schedule for the availability of this software is still undecided.