MIDI Polyphonic Expression

Version 1.0
March-12-2018

Published By:
The MIDI Manufacturers Association
Los Angeles, CA

RP-053 MIDI Polyphonic Expression

MMA
POB 3173
La Habra CA 90632-3173 USA

Table of Contents

1. Introduction

1.1 Background

This specification is designed for MIDI controllers that allow the performer to vary the pitch and timbre of individual notes while playing polyphonically. In many of these controllers, pitch is expressed by lateral motion on a continuous playing surface, while individual timbre changes are expressed by varying pressure, or moving fingers towards and away from the player.

MPE specifies the MIDI messages used for these three dimensions of control — regardless of how a particular controller physically expresses them — and defines how to configure devices to send and receive this “multidimensional control data” for maximum interoperability. Additional dimensions of control may be defined in the future.

MIDI Pitch Bend and Control Change messages are Channel Messages, meaning they affect all notes assigned to that Channel. To apply Channel Messages to individual notes, an MPE controller assigns each note its own Channel. The MIDI 1.0 Specification already includes a configuration option (a Channel Mode Message) for placing each note on its own Channel called “Omni Off / Mono” (MIDI Mode 4, aka “Mono Mode”). Mono Mode is not the preferred operating mode for MPE, because when every Channel must be monophonic, polyphony is limited to 16 (or fewer) notes.

The MPE Specification defines how to perform per-note control on polyphonic (MIDI Mode 3, “Poly Mode”) Channels, but will also work with synthesizers that support Mono Mode, with some restrictions.

1.2 Overview

The MPE specification defines a MMA/AMEI Recommended Practice for hardware and software manufacturers to communicate multidimensional control data between MIDI controller instruments, synthesizers, digital audio workstations, and other products, using MIDI 1.0 messages.

The specification describes a recommended way of using individual MIDI Channels to achieve per-note control without requiring Mono Mode (which would restrict polyphony to a maximum of 16 notes). This enables richer communication between increasingly expressive MIDI hardware and software.

MPE achieves per-note control in a way that is similar to MIDI Mode 4 (“Mono Mode”):

- Wherever possible, every sounding note is assigned its own MIDI Channel for the lifetime of that note. This allows Control Change and Pitch Bend messages to be addressed uniquely to that note.
- A Registered Parameter Number is used to set the range of Channels over which notes are sent and received. The MIDI Channel space can be divided into sub-spaces called Zones, so that multi-timbral playing is still possible using only one MIDI cable.
- Each Zone has a dedicated extra Channel, called the Master Channel, which conveys information common to all notes in that Zone, including pedal data and overall Pitch Bend.

However, MPE addresses the need for higher polyphony by operating in MIDI Mode 3 (“Poly Mode”) where every Channel can be polyphonic. If the number of active notes exceeds the number of available Channels, two or more notes will have to share a Channel. Any notes that share a Channel will not be uniquely controllable, but this can be preferable to limiting polyphony by preventing a new note from sounding, or stopping an older note.

MPE also defines these additional behaviors for senders and receivers:

- Pitch Bend is, by default, set to a range of ±48 semitones for per-note bend and ±2 semitones for Master bend. Either range may be changed to a number of semitones between 0 and ±96 using RPN 0.
- Aftertouch is sent using the Channel Pressure message. To preserve compatibility with existing MIDI devices, Polyphonic Key Pressure may be used with notes on the Master Channel, but not on other Channels.
- In addition to being able to express per-note pitch (Pitch Bend) and pressure (Channel Pressure), a third dimension of per-note control may be expressed using MIDI CC #74.

An implementation summary is tabulated in the Appendix.

1.3 Terminology

Active Note. Any note for which a Note On message has been delivered, but a Note Off message has not.

Lower Zone. The Zone that encompasses Master Channel 1, and is allocated MIDI Channels increasing from Channel 2.

Member Channel. Any MIDI Channel within a Channel Zone that is not a Master Channel.

Occupied Channel. A Channel with at least one active note.

Released Note. A note for which a Note Off message has been delivered. A Released Note may continue to sound for considerable time, most often owing to the length of a release envelope or an interaction with the sustain or sostenuto pedal.

Master Channel. A MIDI Channel reserved for conveying messages that apply to the entire Zone.

Upper Zone. The Zone that encompasses Master Channel 16, and is allocated MIDI Channels decreasing from Channel 15.

Zone. A group of contiguous MIDI Channels comprising a Master Channel and one or more Member Channels.

2. Detailed Specification

2.1 MPE Mode Configuration

The MPE mode of operation (“MPE Mode”) is enabled in a controller or a synthesizer when at least one MPE Zone is configured.

All MPE-compatible devices must support the MPE Configuration Message, in addition to any optional means (such as a power-up default, or via on-board selection) for configuring MPE Mode. The MPE Configuration message is typically received from a digital audio workstation or a remote control application.

MPE Mode is deactivated by setting both Zones to use no Channels. The behavior of a device when MPE operation is deactivated is left to the manufacturer.

2.1.1 MPE Configuration Message (MCM)

The MPE Configuration message (AMEI/MMA CA-034) is defined as Registered Parameter Number "00 06". The MSB of Data Entry represents the number of MIDI Channels assigned, as explained below. The LSB of Data Entry is not used.

[REGISTERED PARAMETER NUMBER]

```
MSB   LSB         Function
=======================================
00    06          MPE Configuration RPN

Message Format: [Bn 64 06] [Bn 65 00] [Bn 06 <mm>]

      n =  MIDI Channel Number

           n=0: Lower Zone Master Channel
           n=F: Upper Zone Master Channel
           All other values are invalid and should be ignored

      mm = Number of Member MIDI Channels in the Zone

           mm=0:       MPE is Off (No Channels)
           mm=1 to F:  Assigns that number of MIDI Channels to the Zone
                       (see below)
```

Each MPE Configuration Message (MCM) defines one MPE Zone, which is determined by the MIDI Channel Number (“n”) nibble of the Message. There can be either one or two zones on MPE devices: a “Lower Zone” and/or an “Upper Zone”. This terminology suggests the lower and upper sections of a split keyboard, but the Zones may map to a single physical controller in many conceivable ways, the details of which are left to the manufacturer. The MCM assigns the number of MIDI Channels to be used for MPE Mode operation in that Zone. Any MIDI Channels not assigned to any Zone remain available for conventional use. It is necessary to send only one MCM if a device intends to use only one Zone.

The Lower Zone is controlled by Master Channel 1, with Member Channels assigned sequentially from Channel 2 upwards. The Upper Zone is controlled by Master Channel 16, with Member Channels assigned sequentially from Channel 15 downwards. Each Zone is activated with its own message, which can be sent in either order. Sending an MCM with the number of Member Channels set to zero deactivates that Zone.

The Master Channel of an unused Zone can be used as a Member Channel for the other Zone. Thus, if only one Zone is active, it may use up to 15 Member Channels (“mm” = F).

A MIDI Channel cannot be assigned to more than one Zone at a time so, in the case where an MCM configures a Zone to include MIDI Channels that were previously assigned to another Zone, the most recent message takes precedence (those MIDI Channels are reassigned to the newer Zone), even if this results in deactivating a Zone.

2.1.2 MCM Examples

One MPE Configuration Message is used to configure an MPE Zone, and two for two Zones. The following figures illustrate some assignments of MIDI Channels as Master Channels and Member Channels in Zones. The Master Channel is illustrated with hatching. (If using Running Status, omit bytes in parenthesis.)

2.1.2.1 Example One

Enable the Lower Zone using 7 Member Channels (2–8), and turn off the Upper Zone.

```
B0   79   00   (B0) 64   06   (B0) 65   00   (B0) 06   07
BF   79   00   (BF) 64   06   (BF) 65   00   (BF) 06   00
```

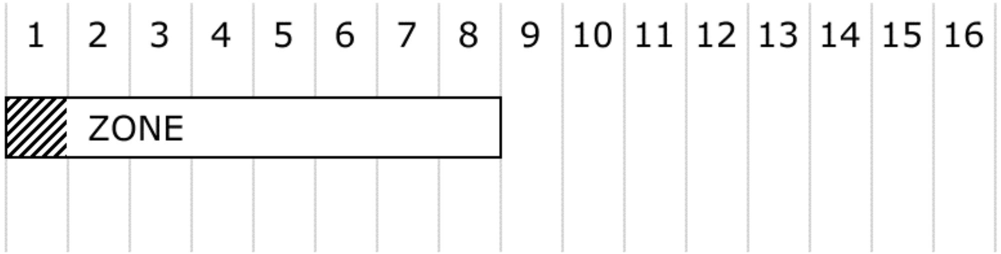

Fig. 1.
Single MPE Zone:
Master Channel 1, with 7 Channels (2–8)

2.1.2.2 Example Two

Enable the Lower Zone using 7 Channels (2–8), and the Upper Zone using 7 Channels (9–15).

```
B0   79   00   (B0) 64   06   (B0) 65   00   (B0) 06   07
BF   79   00   (BF) 64   06   (BF) 65   00   (BF) 06   07
```

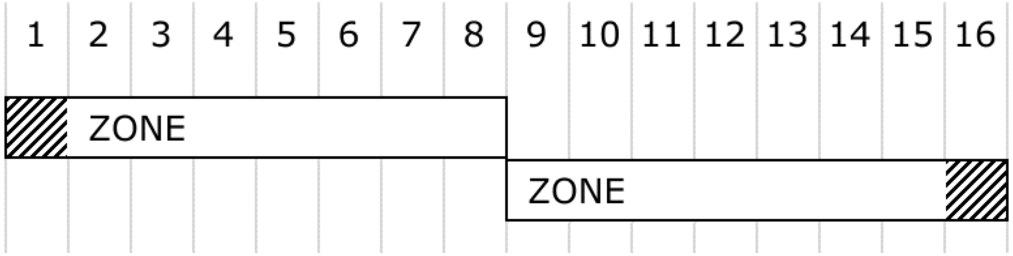

Fig. 2.
Two MPE Zones:
Master Channel 1, with 7 Channels (2–8);
Master Channel 16, with 7 Channels (9–15)

2.1.2.3 Example Three

Enable the Upper Zone using 15 Member Channels (1–15).

Because the Lower Zone is not allocated, Channel 1 is used as a Member Channel for the Upper Zone.

```
BF   79   00   (BF) 64   06   (BF) 65   00   (BF) 06   0F
```

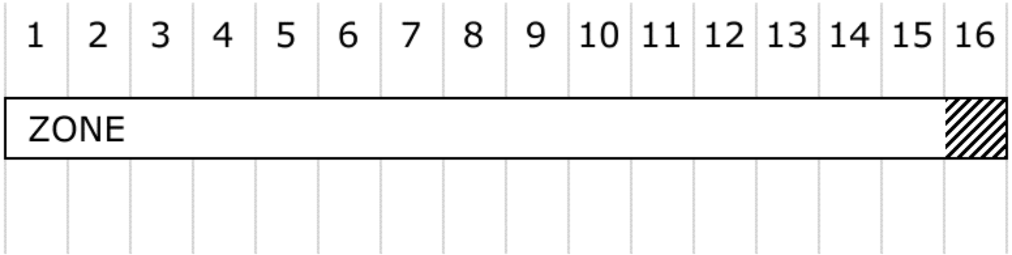

Fig. 3.
Single MPE Zone:
Master Channel 16, with 15 Channels (1–15)

2.1.3 Power-On Default Behavior

At the manufacturer’s discretion, a device may be configured to enter MPE Mode on power-up. When powered on for the first time, it is recommended that compatible devices will send and receive note data on the Lower Zone, using Member Channels 2-16 in Poly Mode [MIDI Mode 3]. This would provide a good initial experience in a monotimbral mode and, in the case of accidental misconfiguration, would be the ‘least wrong’ thing to do.

2.1.4 Receiver Behavior when Resetting Zones

To avoid the possibility of a transmitter leaving a receiver with hanging notes when changing Zone configurations, receivers are required to stop all ongoing notes and reset all controls to reasonable default values on each Channel entering or leaving MPE control.

2.2 Channel and Voice Assignment

2.2.1 MIDI Mode 3 (“Poly Mode”)

The usual mode of operation for polyphonic MIDI instruments is MIDI Mode 3 (Omni Off, Poly). In this Mode, a Channel is maximally polyphonic: it will handle as many simultaneous notes as possible. This is the mode of operation for which MPE is designed.

An MPE controller assigns every new note its own MIDI Channel, until there are no unoccupied Channels available. An occupied Channel becomes unoccupied when its sounding notes have all sent or received Note Off messages.

When there are more Notes than unoccupied Channels, a new note must share a MIDI Channel with an existing note. Since Control Change and Pitch Bend are Channel Messages, they then affect both notes on that Channel.

[Note: Recommendations about the ordering of note and control messages in MPE that help to improve compatibility, editability, and the quality of rendered sound are presented in Section 3.3]

2.2.2 MIDI Mode 4 (“Mono Mode”)

MPE is principally intended to extend the capabilities of MIDI Mode 3 (Omni Off, Poly). However, MPE may also be used with controllers and synthesizers that support MIDI Mode 4 (Omni Off, Mono). A Channel configured for MIDI Mode 4 is monophonic. Starting a note in such a Channel when one is already playing will stop the older note, possibly invoking a legato transition between the old and new notes. This mode is thus ideal for controllers that model string instruments, in which a ‘one Channel per string’ allocation assists realistic rendering of hammer-on and pull-off. Mode 4 is also suitable for controlling a bank of monophonic synthesizers.

A device that is configured in Mono Mode will have one or more Channels set to operate in MIDI Mode 4, and one additional Channel immediately below that which may be used as the Global Controller Channel. This Channel sends or receives instructions that apply to all of the Mode 4 Channels, such as Program Change and Damper Pedal messages. This setup is very similar to the Lower Zone configuration in MPE. If the Channel ranges all match, then the MIDI Mode 4 device will behave identically whether or not it supports MPE.

When using the Upper Zone in Mode 4, however, MPE’s Master Channel will be Channel 16, whereas the Global Controller Channel for MIDI Mode 4 will be different. Devices that are MPE-aware will therefore behave differently from those that are not. It is therefore recommended to use the Lower Zone for control under MIDI Mode 4.

2.2.3 Switching MIDI Modes

If an MPE-compatible synthesizer supports both MIDI Mode 3 and Mode 4, it must switch between them in accordance with the MIDI 1.0 Specification, upon reception of the appropriate MIDI Mode message — CC #126 or #127 — on its “Basic Channel” (the lowest-numbered Member Channel).

2.3 Note Level Messages versus Zone Messages

2.3.1 Messages that are Zone Messages and not Note Level Messages

An MPE Zone normally represents one polyphonic instrument in which certain MIDI messages (for example, Damper Pedal) can be expected to affect all sounding notes across all Member Channels. To reduce MIDI traffic and make event editing easier, those messages should be sent only on a Zone’s Master Channel (not on Member Channels). If an MPE synthesizer receives one of those messages on a Member Channel, it must ignore it. See Table 1 for other MIDI messages that are Zone Messages but not Note Level Messages.

2.3.2 Messages that are both Zone Messages and Note Level Messages

Some other MIDI messages (for example, Pitch Bend) are handled differently. The Pitch Bend wheel on a typical MIDI controller affects all sounding notes, which makes it a Zone Message. But because MPE also offers per-note Pitch Bend (using the same MIDI message), Pitch Bend is both a Zone Message and a Note Level Message. If an MPE synthesizer receives Pitch Bend (for example) on both a Master and a Member Channel, it must combine the data meaningfully. The same is true for Channel Pressure. See Table 1 for other MIDI messages that are both Zone and Note Level Messages.

2.3.3 Messages with defined special behavior

Program Change is a special case. In MIDI Mode 3 it is applied only at the Zone Level to enable monotimbral playing across an entire Zone. In MIDI Mode 4, however, synthesizers may accommodate different programs on different MIDI Channels within the Zone, to allow controllers that imitate string instruments to affiliate programs with each individual string. Therefore, Program Change messages may be sent on Member Channels when a device is operating in MIDI Mode 4, and a receiver operating in MIDI Mode 4 should apply Program Change messages received on Master Channels and Member Channels in the order in which they are received. But a receiver operating in Mode 3 should ignore Program Change messages received on Member Channels. See Table 1 for other MIDI messages that are special cases.

Table 1.
How an MPE Device Uses Common MIDI Messages

	Use at Note Level?	Use at Zone Level?
Pitch Bend, Channel Pressure, CC #74 *(First three dimensions of per-note control)*	Send: Yes[1] Receive: Yes[2]	Send: Yes[1] Receive: Yes[2]
Pitch Bend Sensitivity [RPN 0]	Send: Yes[3] Receive: Yes	Send: Yes Receive: Yes
Polyphonic Key Pressure	No	See Section 2.5
Program Change *(Includes Bank Select CC #0 and #32)*	Valid in MIDI Mode 4 only.	Yes
Reset All Controllers (CC #127)	No	Yes
MIDI Mode Messages[4]	Lowest Member Channel Only (Basic Channel).	No
MPE Configuration Message [RPN 6]	No	See Section 2.1

All other Control Change messages, for example:
CC #1 and #33 [Modulation];
CC #7 and #39 [Volume];
CC #64 [Damper Pedal];
CC #120 [All Sounds Off];
All other RPN messages
All NRPN Messages
	Send: Not recommended.
	Receive: Cannot be expected to respond.
		Yes

Note On/Off messages	Yes	See Section 3.2
System Common, System Real-time, and System Exclusive	Affect the entire system	

Notes:

[1] MPE Controllers may suspend sending these messages when desired. There is no requirement that an MPE Controller be capable of sending all of these messages, but it must be capable of sending at least one dimension of per note control when desired.

[2] MPE Receivers are **required** to respond to these messages on Master and Member Channels.

[3] Send to every Member Channel

[4] Only Modes 3 and 4 are supported. (CC #124 and 125 should not be sent and will be ignored.)

2.4 Pitch Bend and Pitch Bend Sensitivity

MPE allows for Pitch Bend at the Note and Zone levels using the MIDI Pitch Bend message. At the Zone level, Pitch Bend is typically achieved through movement of a control (for example, a pitch wheel or a tremolo bar). At the Note level, Pitch Bend is typically achieved by the movement of a single finger on the playing surface.

When a receiver receives an MPE Configuration Message, it must set the Master Pitch Bend Sensitivity to ±2 semitones, and the Pitch Bend Sensitivity of the Member Channels to ±48 semitones. The values may then be changed using Registered Parameter Number [RPN] 0, in accordance with the MIDI 1.0 Specification. Because the Zone Pitch Bend Sensitivity is controllable independently from that of the Member Channels, setting them is accomplished as follows:

- Master Pitch Bend Sensitivity is set by sending RPN 0 to the Master Channel.
- Pitch Bend Sensitivity on the Member Channels is set by sending RPN 0 to every Member Channel individually.

Channels within the same Zone are not permitted to have different Pitch Bend Sensitivity values. A receiver must apply the last Pitch Bend Sensitivity message received on any Member Channel to all Member Channels in the Zone. To set the Pitch Bend Sensitivity of Member Channels, we recommend that the Pitch Bend Sensitivity message be sent individually to every Member Channel to improve compatibility with all MIDI devices.

The use of RPN 0 presents the option of supplying a less significant byte (LSB), for conveying the microtonal fractions of Pitch Bend Sensitivity. It is recommended that MPE devices use an integer number of semitones for the range and either transmit the LSB as zero, or not transmit it at all so that the receiver infers zero. MPE devices can still choose to respond to 14-bit Pitch Bend Sensitivity messages for compatibility with other equipment.

To simplify interface design, MPE devices can limit their communication to a whole number of semitones between 0 and ±96. (At ±96 semitone resolution, the granularity of 14-bit Pitch Bend data is still smaller than 1.2 cents.) All MPE receivers are required to respond to Pitch Bend at the Zone and Note level. If a device receives Pitch Bend on both a Master Channel and Member Channel, it must combine such data meaningfully and separately for each sounding note.

2.4.1 Calibrated Pitch Instruments

Some MIDI instruments have a pitch control that is absolutely calibrated. These permit the user to bend precise fractions of a scale, as if moving along the fingerboard of a string instrument. Such devices require an invariant mapping to be maintained between 14-bit MIDI Pitch Bend messages and physical space.

Most such controllers will change their sensitivity in response to a Pitch Bend Sensitivity message. They may then attempt to relay the Pitch Bend Sensitivity message to all connected devices. When such a controller’s Pitch Bend Sensitivity is set to zero, it should stop modulating pitch. Setting the sensitivity to zero will configure the controller to operate in a different performance mode: for example, to render a long slide as a sequence of separate notes instead of Pitch Bend data.

2.5 Channel Pressure and Polyphonic Key Pressure

MPE uses MIDI Channel Pressure messages at both the Note and Zone levels to convey pressure. All MPE receivers are required to respond to Channel Pressure at the Note and Zone level. If a device receives Channel Pressure on both a Master Channel and Member Channel it must combine such data meaningfully and separately for each sounding note.

Polyphonic Key Pressure must not be sent on Member Channels. It is currently reserved for a future extension of this specification. Polyphonic Key Pressure may be sent for notes on the Master Channel at the discretion of the implementer, to preserve compatibility with non-MPE-aware devices.

2.6 Third Dimension of Control

In addition to Pitch Bend and Channel Pressure, MPE controllers may provide a third dimension of continuous control. For example, some instruments inspired by the piano keyboard can track finger movement between the front edge of the key and the fallboard. This additional dimension is mapped to Control Change #74.

All MPE receivers are required to respond to CC #74 at the Zone and Note level. If a device receives CC #74 on both a Master Channel and Member Channel, it must combine such data meaningfully and separately for each sounding note.

2.7 Further Dimensions of Control

Control Change messages for additional dimensions of control may be defined in the future.

3. Implementation Guidelines for Improved Compatibility

3.1 Use of One MPE Zone

Although MPE supports the creation of two Zones, many controllers and synthesizers will support only one. Whenever two Zones are defined, such devices can use only the Lower Zone, or may provide a way for the user to choose which Zone to use.

Where interface design permits, an instrument or controller should be able to display the currently selected Master Channel and the range of Member Channels.

3.2 Allocation of Notes to Member Channels

Simple circular assignment of new notes to Member Channels of a Zone will not usually provide satisfactory results. In the simplest workable implementation, a new note will be assigned to the Channel with the lowest count of active notes. Then, all else being equal, the Channel with the oldest last Note Off would be preferred. This set of rules has at least one working real-world implementation.

MPE controllers should preferentially re-use a Channel that has been most recently deployed to play a certain Note Number once the previous note has entered its Note Off state. This avoids stacking and chorusing identical notes, which sounds bad in monotimbral applications, and affects synthesizers that are not designed specifically for MPE.

However, in particular circumstances it is appropriate to have the same Note Number active on two different MIDI Channels. For example, a note may start at a certain pitch and be bent to another before a second note is initiated at the original pitch. Alternatively, a guitar-type controller might permit the same pitch to be played simultaneously on different strings.

When all Channels are occupied by active notes, a controller may choose the Channel in which the change of pitch for the new note requires the smallest adjustment of pitch for other playing notes. Alternatively, at least one commercial implementation provides gentle degradation of pitch control when all Channels are occupied by switching to a mode where notes step discretely from one pitch to the next, permitting Pitch Bend to respond only to small vibrato gestures.

For the sake of MIDI 1.0 compatibility, Note On/Off messages are permitted on the Master Channel, and a synthesizer must respond to these. The practical limitation is that Channel Pressure, Pitch Bend, and Modulation cannot be controlled for notes on a Master Channel without also affecting the rest of the Zone.

3.3 Maximizing Compatibility and Sound Quality Under MPE

The transition to MPE Mode is intended to disrupt users as little as possible. Devices that are wholly MPE-compatible will behave only slightly differently from those that are not. Many users will neither notice nor care about these differences, but more sophisticated hardware and software may need a dedicated “MPE Mode” to adapt to the new behavior.

The ramifications for designers of more sophisticated software can be fairly demanding, particularly where MIDI data can be edited, merged, and looped. Making the MPE workflow transparent presents two challenges:

1. **Software must elegantly provide note editing across Channels without troubling the user.**

Notes may be moved and inserted, and MIDI streams may be merged. To make this easier in Poly Mode, originating Channel numbers do not have to be preserved during editing. Member Channels may be dynamically reassigned by software during playback or retransmission.

Mono Mode and standard MIDI behavior still requires preservation of Channel numbers. From a programmer’s perspective, this entails a far more sophisticated note model. A note is no longer just a pair of timestamped Note On and Note Off messages: it must become an entity with its own timeline of multidimensional control data that can be moved across the time and channel spaces along with the note. This leads to a second challenge, regarding MPE’s implications for user interaction:

2. **The state-affecting behavior of Pitch Bend, which is usually treated as a special case by editing software, now applies across other dimensions of control.**

The pitch of a new note is influenced by the Pitch Bend message most recently received on its Channel before Note On, so a synthesizer must continue to track it even when no note is playing. Differences here between MPE and conventional MIDI practice are:

- Master Channel Pitch Bend also influences the note’s initial pitch.
- The note will cease to be affected by Pitch Bend messages on its Channel after the Note Off message occurs.

In MPE, Channel Pressure and all other per-note control messages work this way too. Values for all these must be tracked and stored on all Member Channels, even when no note is playing, to provide an initial state for a new note.

The prevention of per-note control after Note Off allows rapid reuse of unoccupied Channels, and applies even to notes that are kept active by a Damper Pedal message or a long release envelope.

3.3.1 Note On Setup Example

To play a note that sounds one quartertone above middle C with an initial timbre value of 64, the messages may be ordered as follows (using MIDI Channel 3 as an example):

```
MIDI Message   Description          Effect
================================================================================
E2  2B  41     Pitch Bend           Quarter-tone bend upwards, assuming
                                    sensitivity is set to 48 semitones
B2  4A  40     Controller Change    CC #74  = 40 (64 decimal)
D2  00         Channel Pressure     Set to zero (see Section 3.3.4)
92  3C  38     Note On              Note = Middle C; Velocity = 38 (56 decimal)
```

Provided that the Note On follows all necessary initial settings for pitch and articulation, other orderings of these messages will work equally well. The device that sends these messages may omit any dimensions of control that it does not support, so special care must be taken when merging MPE streams from different instruments.

This is recommended practice because it prevents ‘swooping’ noises when a controller is set to an arbitrary initial value and corrected only after the note has started playing. However, it will also affect any other notes that are being played on the Channel.

3.3.2 Note Off Velocity

Some controllers and synthesizers use Note Off velocity to convey meaningful information. This parameter is part of the MIDI 1.0 Specification, but is not universally adopted. When receiving a Note Off message sent using the MIDI shorthand of ‘Note On with velocity zero’, it is recommended that this message be interpreted as Note Off velocity 64. This is a reasonable compromise to improve compatibility with synthesizers that respond to Note Off velocity.

3.3.3 Pitch Bend

Under MPE, control of a note ceases once Note Off has occurred. As a consequence, any feature which requires continual transmission of Pitch Bend for the purposes of automation, such as a controller’s automatic pitch quantization, must be designed to have no effect once a note has been released. Fig. 6 shows a typical timeline of a non-Master Channel handling more than one note.

The behavior of Master Pitch Bend is unaltered. It applies to every note within the Zone that is still sounding: even those that have passed into their Note Off phase and are sustained only by a pedal or a release envelope.

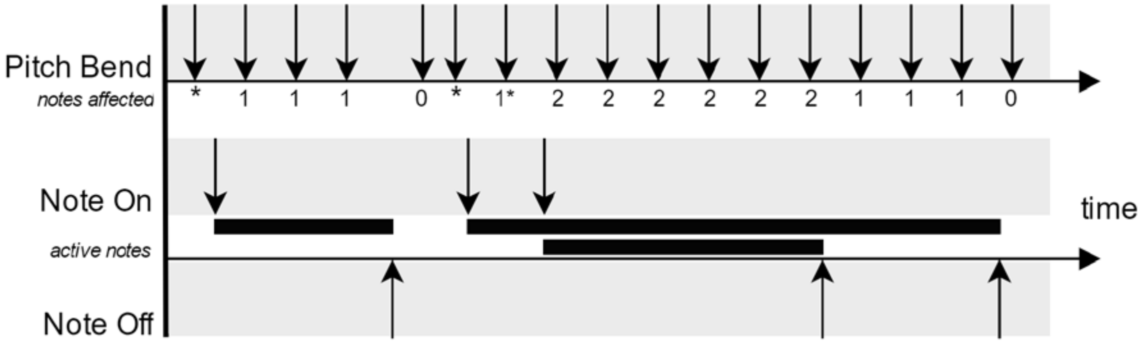

Fig.6.
Example timeline for a Channel with three active notes.
This behavior also applies to all other dimensions of control.
‘*’ under ‘notes affected’ signifies that the value sets the initial pitch of the next note.

3.3.4 Channel Pressure

Many synthesizers are designed for use with controllers that set Channel Pressure to zero before the termination of a note. For full compatibility with these, and to simplify sound design on MPE-compatible synthesizers, Channel Pressure must be set to zero immediately before a Note On or a Note Off wherever it is appropriate to the design of a controller.

Not all controllers can be expected to behave in this way. In the simulation of certain hand drums, for example, pressure applied to the drum skin is adjustable independently from the note creation mechanism.

3.3.5 Timbre Control

If a device offers a third dimension of control it will use Control Change #74 and typically control timbre. Broadly, there are two schemes for this transmitting this dimension from a control instrument:

> **Initial-position.** The value of CC #74 at Note On encodes the initial position of a user's interaction with the instrument. An example of an *Initial-position* controller would be a digital hi-hat, in which CC #74 might encode the radius of the striking position, from bell to rim.
>
> **Initial-64.** The control pertains to a dimension that may be varied once a note has been struck. The initial position of CC #74 under such circumstances must be 40h (64 decimal), such that movement can follow in either a positive or negative direction.

Sometimes an instrument is sensitive both to initial position and subsequent movement along the third axis. An example of this would be a digital piano that can sense the player's finger position between the front edge of the key and the fallboard when a key is depressed, and continues to track this throughout the note's duration. Under such circumstances, neither scheme for representing timbre will fit every possible mode of interaction with a synthesizer.

Sequencers, digital audio workstations, and arpeggiators are examples of editing software that allow a note's start point or duration to be altered, or continuous data within the duration of the note to be extended or deleted. Such software introduces complications when working with MPE. Adjustments can have an unintended effect on data within the edited note. The software designer is then left with the difficulty of determining the musical intention behind a user's action.

An instrument may do any of the following to assist with these difficulties:

1. Allow the player to choose between the *Initial-position* and *Initial-64* schemes of interaction on the instrument itself, changing the way in which data is generated by the instrument.
2. Fix CC #74 to *Initial-64* behavior, and designate another CC to convey the static initial position. One such implementation employs CC #75 for initial position, which simplifies the sound designer's appropriation of CC #74.
3. Fix CC #74 to *Initial-position* behavior, and leave the MIDI transformation and manipulation questions to whatever is handling the data.

Either or both of the first two actions are preferable to the third.

3.4 MPE Plugin Examples

Examples of the following feature implementations are presented in the Appendix.

- A VST plugin may signal that it is MPE compliant by implementing the canDo method that returns a non-zero value for the string "MPE".
- An Audio Unit plugin must support a new type of property.

4. Appendix

Implementation Summary Table

Action	Method
Select MPE mode and assign Channels	Send up to two RPN 6 messages, on Channel 1 and Channel 16, to allocate Channels to the Lower and Upper Zones.
Set the Pitch Bend Sensitivity for a Master (Zone) Channel	Send a RPN 0 message (whole number of semitones recommended, accepted range ±1 to ±96 semitones) on the Master Channel. Default is ±2 semitones.
Set the Pitch Bend Sensitivity for Member (Note) Channels	Send individual RPN 0 messages (whole number of semitones recommended, accepted range ±1 to ±96 semitones) to every Member Channel. Default is ±48 semitones.
Pitch Bend, for a Zone	Send a MIDI Pitch Bend message to the Master Channel.
Pitch Bend, for a note	Send a MIDI Pitch Bend message to the note’s Channel.
Pressure/Aftertouch, for a Zone	Send a Channel Pressure message to the Master Channel. *Polyphonic Key Pressure may be recognized on the Master Channel for compatibility with existing MIDI modes.*
Pressure/Aftertouch, for a note	Send Channel Pressure message to the note’s Channel. *Do not send Polyphonic Key Pressure on a Member Channel.*
Third dimension, for a Zone *Usually mapped to timbre*	Send Control Change #74 to the Master Channel.
Third dimension, for a note *Usually mapped to timbre*	Send Control Change #74 to the note’s Channel.
Pedals, Program Change, Reset All Controllers, and other Zone data	Send the appropriate message on the Master Channel of the affected Zone.

VST canDo Implementation

```
/** Example of querying a VST 2.4 plug-in for MPE support. **/
VstInt32 MyVstplugin::canDo (char* text)
{
   if (! strcmp (text, "MPE"))
      return 1;   // 1 = plug-in supports MPE mode

   ...
}

/** Example how to query a VST 2.4 plug-in
    for which CC controllers it supports
    on a note-per-Channel basis, and their labels.
**/
VstInt32 MyVstplugin::vendorSpecific (VstInt32 lArg1, VstInt32 lArg2, void* ptrArg, float floatArg)
{
   if (lArg1 == "MPE")  // literal for MPE Controller
   {
      if (lArg2 == 7)
      {
         strcpy (static_cast<char*>(ptrArg), "Gain");
         return 1;   // 1 = CC supported for MPE
      }
      if (lArg2 == 10)
      {
         strcpy (static_cast<char*>(ptrArg), "Pan");
         return 1;
      }
      if (lArg2 == 74)
      {
         strcpy (static_cast<char*>(ptrArg), "Timbre");
         return 1;
      }

      return 0;    // 0 = CC not supported for MPE
   }

   ...
}
```

Audio Unit Property Implementation

```
/** Example of querying an Audio Unit plugin for MPE support.
    kAudioUnitProperty_SupportsMPE is a built-in constant from iOS 10.10 and OS X 10.12.
**/

bool supportsMPE() const
{
   UInt32 dataSize = 0;
   Boolean isWritable = false;

   if (AudioUnitGetPropertyInfo (audioUnit, kAudioUnitProperty_SupportsMPE,
                                 kAudioUnitScope_Global, 0, &dataSize, &isWritable) == noErr
       && dataSize == sizeof (UInt32))
   {
       UInt32 result = 0;
      if (AudioUnitGetProperty (audioUnit, kAudioUnitProperty_SupportsMPE,
                                kAudioUnitScope_Global, 0, &result, &dataSize) == noErr)
      {
         return result > 0;
      }
   }

   return false;
}
```