

**UNIVERSIDAD NACIONAL DE INGENIERIA**  
**FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**



MIDI, LA INTERFASE DIGITAL PARA INSTRUMENTOS MUSICALES.  
EXPLICACIÓN, DEMOSTRACIÓN E IMPLEMENTACIÓN DE UN SISTEMA  
BÁSICO

***T E S I S***

**PARA OPTAR EL TITULO PROFESIONAL DE:  
INGENIERO ELECTRONICO**

**Armando Esteban Patroni Chávez**

***Promoción 1,993 - 2***

**LIMA - PERU**

**1997**

A mis padres, hermanas  
y a mi familia en general.

A Marcial, mi maestro,  
por sus enseñanzas y su paciencia

A mis compañeros músicos que  
siempre confiaron en mí.

**MIDI, LA INTERFASE DIGITAL PARA INSTRUMENTOS MUSICALES.  
EXPLICACIÓN, DEMOSTRACIÓN E IMPLEMENTACIÓN DE UN SISTEMA  
BÁSICO**

**Título:** MIDI, la Interfase Digital para Instrumentos Musicales. Explicación, demostración e implementación de un sistema básico

**Autor:** Bach. Armando Esteban Pattroni Chávez.

**Grado al que opta:** Ingeniero Electrónico.

**Facultad:** Facultad de Ingeniería Eléctrica y Electrónica.

**Universidad:** Universidad Nacional de Ingeniería.

Lima, 1995

#### **EXTRACTO:**

El presente trabajo consta de cuatro (4) capítulos, el primero de ellos se titula Introducción y se refiere a la presentación de la tecnología objeto de este estudio, sus aplicaciones principales, usos caseros y profesionales, utilidades prácticas y provecho que se pueda extraer de su uso extendido.

En el segundo capítulo, titulado "La Interfase Digital para Instrumentos Musicales", se presenta un estudio completo del protocolo de comunicaciones que conecta hoy en día a los dispositivos que poseen esta interfase en esta generación o la anterior. El estudio es a nivel digital, se presentan específicamente los bytes que se utiliza para comunicar un instrumento con otro, es decir, un estudio completo de los *mensajes MIDI*: mensajes de Canal-Voz, mensajes de Cambio de Programa, etc.

En el Capítulo 3, se presentan las interconexiones de sistema que se requieren dentro de un sistema MIDI, algo de Hardware para manejo de datos MIDI, así como una breve explicación del uso de la PC y sus aplicaciones en un ambiente MIDI.

En el capítulo cuarto se hace un pequeño resumen y explicación de los métodos empleados para grabar secuencias digitales de datos MIDI en una computadora, procedimiento conocido como *secuenciación*.

Se está incluyendo una serie de manuales y copias de folletos de algunos equipos que tienen que ver con la tecnología MIDI.

conecta hoy en día a los dispositivos que poseen esta interfase en esta generación o la anterior. El estudio es a nivel digital, se presentan específicamente los bytes que se utiliza para comunicar un instrumento con otro, es decir, un estudio completo de los *mensajes MIDI*: mensajes de Canal-Voz, mensajes de Cambio de Programa, etc.

En el Capítulo 3, se presentan las interconexiones de sistema que se requieren dentro de un sistema MIDI, algo de Hardware para manejo de datos MIDI, así como una breve explicación del uso de la PC y sus aplicaciones en un ambiente MIDI.

En el capítulo cuarto se hace un pequeño resumen y explicación de los métodos empleados para grabar secuencias digitales de datos MIDI en una computadora, procedimiento conocido como **secuenciación**.

Se está incluyendo una serie de manuales y copias de folletos de algunos equipos que tienen que ver con la tecnología MIDI.

## TABLA DE CONTENIDOS

Prólogo		1
Capítulo I:		
	INTRODUCCIÓN.	3
1.1	Breve historia.	5
1.2	Producción musical electrónica.	9
1.2.1	Producción musical casera vía MIDI.	10
1.2.2	MIDI en el estudio.	13
1.2.3	Uso de la interfase MIDI en audio para video y películas.	14
1.2.4	MIDI en vivo.	15
1.3	Resumen.	16
Capítulo II:		
	INTERFASE DIGITAL PARA INSTRUMENTOS MUSICALES.	18
2.1	¿Qué es el MIDI?	18
2.1.1	La palabra digital.	19
2.2	El mensaje MIDI.	21
2.2.1	Canales MIDI.	24
2.2.2	Modos MIDI.	25
2.2.3	Mensajes canal-voz.	31
2.2.4	Mensajes canal-modo.	49
2.2.5	Mensajes de sistema.	51
2.2.6	Status corriente.	58

2.3	Filtraje MIDI.	59
2.4	Mapeo MIDI.	60
2.5	Resumen.	61
Capítulo III:		
	SISTEMAS DE HARDWARE DENTRO DE LA PRODUCCIÓN MIDI.	62
3.1.	Interconexión del sistema.	62
3.2	El cable MIDI.	62
3.2.1	Puertos MIDI.	64
3.2.2	Eco MIDI.	66
3.2.3	Configuraciones típicas.	67
3.3	Hardware de manejo de datos.	72
3.3.1	MIDI Merger.	72
3.3.2	MIDI Thru Box.	73
3.3.3	MIDI Switcher.	74
3.3.4	MIDI Patchbay	75
3.3.5	Conversor MIDI/Trigger.	77
3.3.6	Procesadores de datos MIDI.	79
3.3.7	Herramientas de diagnóstico MIDI.	80
3.3.8	La red de área local (LAN).	81
3.4	La computadora personal en la producción MIDI.	83
3.4.1	IBM/Compatible.	84
Capítulo IV:		
	SECUENCIACIÓN.	87
Apéndice A:		
	THE MIDI 1.0 SPECIFICATION	90
Apéndice B:		



THE MIDI IMPLEMENTATION CHART	109
CONCLUSIONES	118
BIBLIOGRAFIA	120

## PROLOGO

A lo largo de la historia, las herramientas usadas por los artistas: pintores, escultores, escritores y músicos, ha experimentado un continuo cambio. Algunos cambios han sido más significativos que otros. Sin embargo todos ellos han mejorado el proceso artístico de una manera u otra.

Al haber trabajado durante la década pasada en actividades musicales, he tenido la oportunidad única de experimentar uno de los cambios más radicales en herramientas de uso artístico que haya ocurrido en la historia de la creación musical. Si la comparamos con el desarrollo del órgano de tubos o la grabación sonora, la integración de las computadoras con los instrumentos musicales ha cambiado una vez más la manera como se hará música de hoy para siempre. El ímpetu detrás de este surgimiento fue el desarrollo en 1983 de la Interfase MIDI. Con esta nueva tecnología los músicos pueden ahora virtualmente interconectar todos los elementos de su estudio y usar la potencia de la computadora para administrar todo este poder que ahora tienen en sus manos.

Aunque fue muy revolucionaria, esta ya no es una tecnología nueva. Se ha vuelto ya un lugar común en el

ambiente musical, lo cual es un gran mérito para una tecnología nueva. Sin embargo, esta tecnología y su desarrollo no se han convertido en tema frecuente en los ambientes técnicos de nuestro país, como debería serlo una nueva forma de aplicar nuestros conocimientos para el desarrollo de alguna actividad externa. La Interfase MIDI sirve para la Música, es la electrónica puesta al servicio del arte, al mismo nivel que la electrónica se puso al servicio del arte al desarrollar la Televisión, la Radio, etc, los artistas de la música se convierten en usuarios de esta nueva tecnología y los Ingenieros Electrónicos son los que desarrollan todas las herramientas necesarias para que el usuario pueda manejar esta novedad sin complicaciones innecesarias; para que el usuario tenga justamente un manejo de instrumentos a nivel de usuario.

Por eso este trabajo ha investigado a fondo el funcionamiento, lenguaje y protocolos de la comunicación MIDI para tener una visión que trascienda ese nivel de usuario y nos dé a los especialistas en electrónica un conocimiento a nivel técnico para poder entender esta nueva tecnología.

La importancia de esta nueva visión del arte también se da a nivel económico, puesto que el MIDI ha permitido que, en un extremo, una persona sola pueda crear una pieza musical sin ayuda alguna, reduciendo enormemente los costos de producción, por tanto incrementando la cantidad de música posible de ser escuchada y, al haber mayor oportunidad de escoger, mejorando en mucho la calidad de la música que se produce hoy en día.

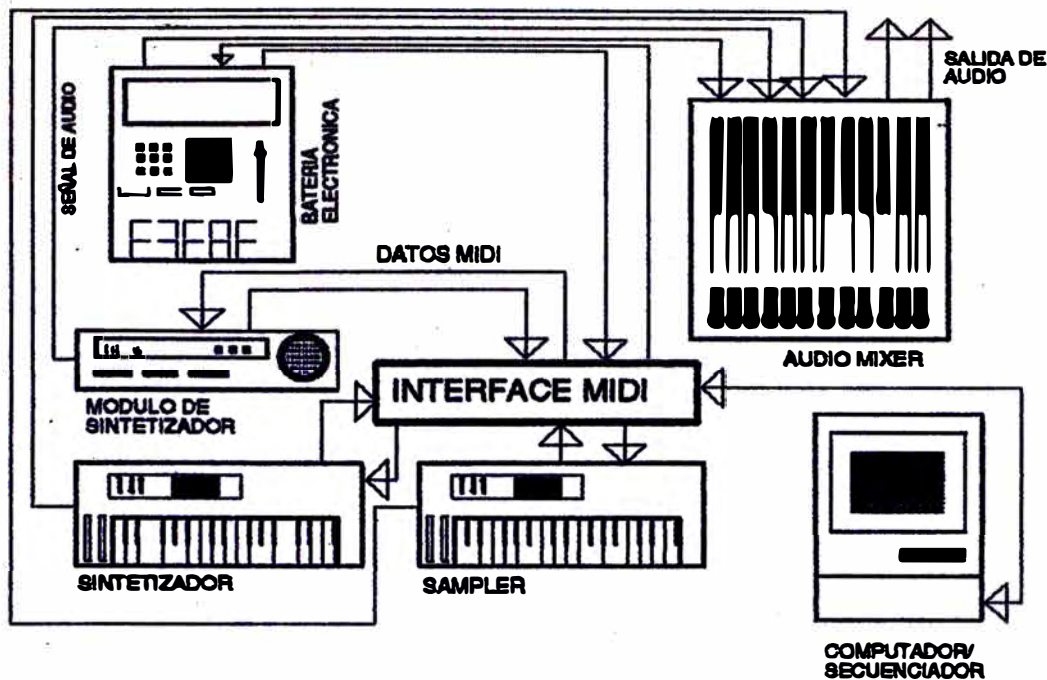
## CAPITULO I INTRODUCCION

El MIDI (Musical Instrument Digital Interface, que en español significa Interfase Digital para Instrumentos Musicales) es un lenguaje digital de comunicaciones y una especificación de hardware compatible que permite comunicar entre sí a múltiples instrumentos electrónicos, controladores de ejecución, computadoras y otros dispositivos relacionados dentro de una red conectada (Fig 1-1). Se usa para traducir eventos de control o ejecución (tal como la acción de tocar un teclado, seleccionar un número de patch, variar una rueda de modulación, etc.) en mensajes digitales equivalentes y transmitir estos mensajes a otros dispositivos MIDI que pueden ser usados para controlar la generación de sonidos y parámetros de tales dispositivos en un esquema de ejecución. Adicionalmente, los datos MIDI se pueden grabar en un dispositivo digital (conocido como secuenciador), que puede usarse para grabar, editar y generar una salida de datos de ejecución MIDI.

En términos musicales, este lenguaje digital es un medio nuevo emergente que permite al músico expresarse con un grado de expresión y control que no era, hasta hace poco, posible a nivel individual. A través de la transmisión de este lenguaje de ejecución digital dentro

de un sistema musical electrónico, un músico puede crear y desarrollar una canción o composición en un ambiente de producción accesible, flexible y práctico. Adicionalmente a la composición y ejecución de una canción, el músico es también capaz de actuar como tecno-conductor, teniendo un control completo sobre un amplio rango de sonidos individuales que son creados por varios instrumentos, su *timbre* (sonido y calidad tonal), y *mixtura* (nivel, panorama y otros controles de tiempo real). El MIDI también puede usarse para variar los parámetros de control de ejecución de los instrumentos electrónicos y dispositivos de efectos durante una ejecución.

El término *interfase* se refiere al enlace y hardware de comunicación de datos que ocurre en una red conectada MIDI. El MIDI hace posible que todos los instrumentos electrónicos y dispositivos sean direccionados en esta red a través de la transmisión de control y ejecución en tiempo real. Es más, cierto número de instrumentos o generadores de sonidos dentro de un instrumento pueden direccionarse independientemente en una línea MIDI simple. Esto es posible pues cada línea MIDI es capaz de transmitir mensajes de control y ejecución sobre 16 canales discretos. Este arreglo multicanal permite al ejecutante grabar, controlar y reproducir datos MIDI en un ambiente de trabajo que asemeja el proceso de grabación multicanal que es tan familiar a la mayoría de los músicos. Al masterizar, sin embargo, el MIDI ofrece desafíos de producción y posibilidades que van más allá



**Fig 1-1. Ejemplo de un típico sistema MIDI**

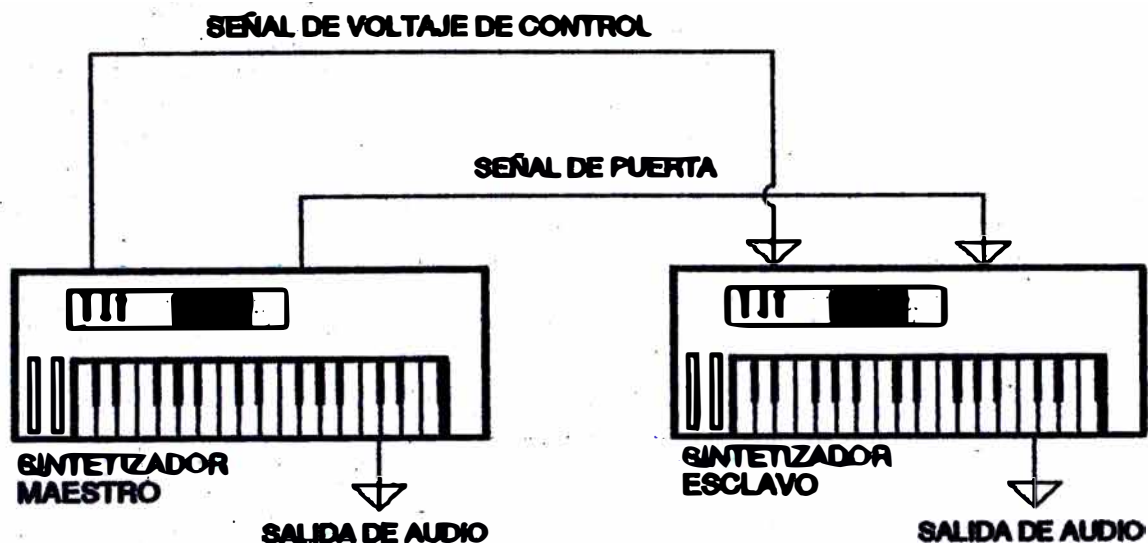
de las capacidades del estudio de grabaciones multicanal tradicional.

### 1.1 Breve historia

En los primeros días de la música electrónica, los teclados sintetizadores eran comúnmente dispositivos *monofónicos* (capaces de hacer sonar sólo una nota a la vez) y frecuentemente generaban una calidad de sonido muy delgada. Estos factores limitantes hicieron que los primeros fabricantes buscaran maneras de interconectar instrumentos para crear una textura de sonido más rica y gruesa. Esto pudo lograrse al establecer un enlace de instrumentos que permitió que un sintetizador (actuando como controlador maestro) sea usado para controlar directamente los parámetros de ejecución de uno o más sintetizadores (conocidos como *módulos de sonido*

*esclavos*). Como resultado, se desarrolló una señal de control conocida como *voltaje de control* o CV (Fig 1-2).

Este sistema estaba basado en el hecho de que en la mayoría de los primeros sintetizadores, al tocar, se generaba un voltaje DC que se usaba para controlar directamente los osciladores controlados por voltaje (que afectaban la afinación de una nota sonora) y los amplificadores controlados por voltaje (que afectaban el volumen de la nota y su naturaleza on/off). En vista de que muchos teclados de aquel tiempo generaban una señal DC que ascendía a razón de 1 voltio por octava (dividiendo cada octava musical en intervalos de 1/12 de voltio) fue posible usar este voltaje de control estandarizado como señal de referencia maestra para transmitir información de afinación a sintetizadores adicionales. Además de un voltaje de control, esta estandarización requería que un teclado transmitiera una *señal de puerta*. Esta segunda señal se usaba para sincronizar el comienzo y los tiempos de duración de cada nota. Con la aparición de *sintetizadores polifónicos* más avanzados (capaces de generar múltiples sonidos a la vez) y los primeros aparatos digitales en el mercado, fue claro que esta estandarización no sería más la respuesta a la necesidad de un control amplio de sistemas, y empezaron a aparecer nuevos estándares en escena (creando estándares de control incompatibles entre si). Con la llegada de las primeras máquinas de ritmos o baterías electrónicas y aparatos de secuencia, la estandarización



**Fig 1-2. Ejemplo de enlace de instrumentos usando voltaje de control y señal de puerta.**

se convirtió en un dilema.

La sincronización entre tales aparatos primitivos era frecuentemente problemática, ya que los fabricantes normalmente estandarizaban sobre diferentes tiempos en el pulso de sincronismo de reloj. Estas incompatibilidades de tiempo causaban pérdida de sincronismo en un período corto de tiempo (cuando se conectaba aparatos de diferentes fabricantes). Esto hacía que la sincronización fuera casi imposible sin el uso adicional de convertidores de tiempo de sincronismo u otros tipos de modificaciones.

Como resultado de estas incompatibilidades, Dave Smith y Chet Wood (entonces trabajando para *Sequential Circuits*, uno de los pioneros en la fabricación de



instrumentos electrónicos) empezaron a crear un protocolo para instrumentos electrónicos digitales, que se denominó la *Interfase Universal para Sintetizadores* (Universal Synthesizer Interface, USI). Como resultado de este protocolo, fue posible que equipos de diferentes fabricantes se comuniquen directamente entre sí. Por ejemplo, el sintetizador de una compañía podría trabajar con el secuenciador de otra compañía. En Octubre de 1981, se propuso la USI a la Audio Engineering Society. En el curso de los siguientes dos años, un panel (que incluía representantes de los más grandes fabricantes de instrumentos electrónicos) modificó este estándar y lo adoptó bajo el nombre de **Musical Instrument Digital Interface** (Especificación MIDI 1.0).

La aceptación del MIDI se debió en gran parte a la fuerte necesidad de un protocolo estandarizado y a los avances en tecnología que permitieron que los chips de complejos circuitos y diseño de hardware sean fabricados con mucha eficiencia de costo. También se debió, en parte, a la introducción del sintetizador Yamaha DX-7 en los primeros meses de 1983, momento a partir del cual las ventas de teclados empezaron a crecer a un ritmo asombroso.

Con la adopción de este estándar industrial, cualquier aparato que incorpore una serie de puertos MIDI en su diseño es capaz de transmitir y/o responder a datos de control y ejecución conforme a la especificación MIDI. Puesto que la vasta mayoría de instrumentos electrónicos

y aparatos implementan este lenguaje de comunicaciones, el músico electrónico puede estar seguro de que las funciones básicas de cualquier nuevo dispositivo se integrarán dentro de un sistema MIDI existente.

### **1.2 Producción musical electrónica**

Hoy en día los sistemas MIDI están siendo usados por muchos músicos profesionales y no-profesionales. A su vez, estos sistemas están tendiendo a desarrollar un número creciente de metas de producción, incluyendo producción musical, audio para pos-producción de video y cine, producción escénica, etc.

Esta aceptación industrial puede atribuirse en gran parte la eficiencia de costo, potencia y velocidad general de las producciones MIDI. Actualmente una gran variedad de instrumentos y dispositivos MIDI accesibles están disponibles en el mercado. Una vez adquiridos, existe frecuentemente menos necesidad (si alguna) de contratar músicos externos para un proyecto de producción. Adicionalmente, el ambiente de producción multicanal MIDI permite al músico componer, editar y arreglar una pieza con un alto grado de flexibilidad, sin la necesidad de grabar y doblar estos sonidos en una cinta multicanal.

Este acceso potencial a una futura expansión y las crecientes capacidades de control sobre un sistema de producción integrado han impulsado el crecimiento de una industria, que es también de naturaleza muy personal. Por primera vez en la historia de la música, es posible que

un individuo realice una producción sonora a escala completa de una manera eficiente en tiempo y costo. Ya que el MIDI es un medio de ejecución en tiempo real, es también posible escuchar y editar completamente esta producción en cada etapa de su desarrollo -todo en la comodidad de su propia casa o estudio de producción personal.

Los sistemas MIDI también pueden diseñarse en cualquier número de configuraciones personales. Pueden diseñarse para adaptarse de la mejor forma a un objetivo de producción (la producción de la pista sonora de un video), al instrumento principal y estilo de ejecución de un artista (percusión, guitarra, teclados) o a los hábitos personales de trabajo de un músico (elección de un equipo o esquema de diseño). Cada una de estas ventajas es un tributo definido a la potencia y flexibilidad que son inherentes a las posibilidades que el MIDI trae a la producción musical moderna.

### **1.2.1 Producción musical casera vía MIDI**

Existe actualmente un amplio rango de instrumentos musicales electrónicos, dispositivos de efectos, sistemas de cómputo y otros dispositivos relacionados al MIDI disponibles en el mercado musical electrónico. Este rango de diversidad permite que el músico electrónico seleccione lo que mejor se adapta a su gusto musical particular y estilo de producción. Con el advenimiento de los *circuitos integrados a gran escala* (Large-Scale Integrated Circuit Chip, LSI) que hace posible que

compleja circuitería sea producida en masa de una forma rápida y fácil, muchos de los dispositivos que forman parte de un sistema musical electrónico están ahora al alcance de casi cualquier músico o compositor, sea este un profesional, aspirante a artista o aficionado. Ya que la mayoría de estos dispositivos son digitales o controlados digitalmente por naturaleza, frecuentemente puede ser simple y eficiente en costo el implementarlos con las capacidades MIDI.

Debido a la naturaleza personal del MIDI, un sistema de producción puede mostrarse en un número de ambientes y puede configurarse para efectuar un amplio rango de aplicaciones. Los sistemas MIDI básicos, y muchos no tan básicos, están siendo actualmente instalados en las casas de un creciente número de músicos electrónicos. Estos van desde sistemas que sólo ocupan una esquina de la habitación del artista a sistemas mayores que han sido instalados dentro de un lugar exclusivo para estudio de producción MIDI.

Los sistemas de producción MIDI caseros pueden diseñarse para servir funcionalmente a un amplio rango de aplicaciones y pueden ser usados por aficionados y aspirantes tanto como por músicos profesionales. Tales sistemas también tienen la importante ventaja de permitir al artista producir su música en un ambiente confortable (en cualquier momento de inspiración). Tales lujos de producción, que hace una década le hubieran costado al artista una fortuna, están ahora al alcance de casi

cualquier músico.

Los sistemas de producción MIDI frecuentemente aparecen en gran variedad de formas y tamaños y pueden diseñarse para adaptarse a un amplio rango de necesidades de presupuesto y producción. Por ejemplo, un instrumento de teclado (conocido comúnmente como *workstation MIDI*, Estación de Trabajo MIDI), es frecuentemente la integración de un teclado, un sintetizador polifónico, sonidos de percusión y secuenciador incorporado, todo en un solo paquete. Este es esencialmente un sistema *todo en uno*, que permite al músico incursionar en la producción de estilo MIDI en forma económica. Se puede agregar fácilmente instrumentos adicionales MIDI a esta workstation simplemente conectando su puerto de salida MIDI OUT al puerto de entrada MIDI IN de el o los nuevos dispositivos. Conecte los canales de audio adicionales a su mezclador y ya está Ud. en el negocio.

Otros sistemas MIDI, hechos de instrumentos y dispositivos discretos, frecuentemente son cuidadosamente seleccionados por el artista para generar un sonido específico o para servir a una función específica dentro del sistema. Aunque este tipo de sistema MIDI no es muy portátil, comúnmente es más poderoso, puesto que cada componente se diseña para ejecutar un rango mayor de funciones de tarea específica. Tal sistema puede incluir uno o más sintetizadores de teclado, módulos de sintetizador, samplers (emuladores), baterías electrónicas, una computadora (con secuenciador y otras

formas de software MIDI), dispositivos de efectos y capacidades de mezcla de audio.

### **1.2.2 MIDI en el estudio**

El MIDI también ha cambiado dramáticamente el sonido, la tecnología y los hábitos de producción dentro de los estudios de grabación profesionales. Antes de la vigencia del MIDI, el estudio de grabación era uno de los únicos ambientes de producción que permitían al artista o compositor el combinar varios instrumentos, estilos y timbres en un producto grabado final. El proceso de grabar un grupo en una puesta en vivo, era y es, frecuentemente, una tarea lenta y cara. Esto se debe al alto costo de contratar músicos de sesión y a la cantidad de horas que se debe pagar por el uso de un estudio de grabación profesional. Además, se gasta un determinado número de horas para capturar la "performance ideal" en la cinta.

Con el advenimiento del MIDI, una gran parte de la producción musical moderna puede ser ejecutada y, en gran parte, producida antes de ir al estudio de grabación. Esta posibilidad ha reducido el número de horas requeridas para colocar las pistas en una cinta multicanal a un mínimo efectivo. Por ejemplo, es ahora un lugar común que los grupos pre-produzcan el proyecto completo de un álbum usando sus propios sistemas MIDI personales. Una vez completado (o casi completado), el grupo puede llevar su sistema MIDI al estudio y colocar los canales de audio aislados en una grabadora

multicanal. Una vez que todas las pistas MIDI han sido transferidas a la cinta, ya puede empezar el proceso de grabar las voces, fuentes en vivo, o emulaciones vocales a la cinta. Posteriormente, la cinta matriz completa puede mezclarse en un producto musical final. Alternativamente, las pistas MIDI pueden no ser grabadas del todo en la cinta multicanal, sino que pueden sincronizarse a la cinta, permitiendo que el secuenciador actúe como una extensión de las pistas grabadas.

### **1.2.3 Uso de la interfase MIDI en audio para video y películas**

La música electrónica también se ha convertido en una gran herramienta para las fases de sonorización y post-producción de audio para películas y video. Las prácticas de producción estandarizada MIDI han permitido que compositores musicales y artistas de efectos para el medio visual se vuelvan cada vez más productivos. Como resultado, el MIDI ha venido a ser reconocido como una herramienta poderosa y eficiente en costo en la producción de comerciales de TV y bandas sonoras completas de películas.

Aunque muchos compositores musicales electrónicos para el medio visual trabajan independientemente, muchos se han asociado con estudios de grabación, casas de producción de video, y firmas audiovisuales para combinar fuerzas a fin de crear un equipo completo de producción musical y visual.

#### 1.2.4 MIDI en vivo

La producción musical electrónica y el MIDI están también firmemente establecidos en la puesta en escena de una performance en vivo. Adicionalmente a la calidad del sonido producido por muchos sistemas modernos de teclado electrónico, la popularidad del MIDI en los escenarios se debe principalmente a dos factores: la programación de la preproducción y la facilidad de controlar fácilmente un número de dispositivos desde una localización central.

La facilidad de secuenciar previamente un número de partes de ritmo y pistas, encadenarlas en una sola secuencia controlable (usando un programa de secuencia como una cajita de música) y tocar esta secuencia en escena se ha convertido en una fuerte herramienta de performance en vivo para muchos músicos. Esta técnica es usada actualmente por artistas solistas que se han vuelto orquestas de un solo músico al adicionar secuencias de pistas (usualmente consistentes en máquina de ritmos y una serie de otros instrumentos) a sus ejecuciones y voces en vivo. Grupos más grandes de techno-pop comúnmente hacen uso extensivo de secuencias en escena para manejar un amplio rango de instrumentos y efectos adicionales a su ejecución en vivo.

Muchos ejecutantes en vivo hacen uso de las capacidades de control en tiempo real del MIDI en una puesta en escena. Esto incluye la facilidad de tocar dispositivos esclavos desde un solo teclado, almohadilla percusiva u otro tipo de controlador. El MIDI también



permite tocar diferentes partes musicales y de percusión desde un solo controlador. Adicionalmente a comunicar datos relacionados a la ejecución, el MIDI es capaz de controlar un número de parámetros de los dispositivos en tiempo real. Por ejemplo, un controlador central MIDI (tal como un teclado controlador o un interruptor de pedal MIDI) puede usarse para cambiar los patches sonoros y parámetros de ejecución (tales como volumen, panorama, etc) de varios instrumentos MIDI o dispositivos de efectos.

El MIDI también está indirectamente involucrado en la creación de muchas performances en vivo. A través de el uso de un programa de notación musical y un sistema de producción, el MIDI puede frecuentemente ayudar a un compositor a ingresar, editar y escuchar una partitura musical. Una vez finalizada, una partitura computarizada de una secuencia o serie de hojas puede imprimirse para distribuirse entre los músicos en escena o en el estudio.

### **1.3 Resumen:**

A partir de estas aplicaciones se puede ver que la Interfase Digital para Instrumentos Musicales (MIDI) se ha tornado ampliamente aceptada por consumidores y profesionales en muchas facetas de la industria musical. También ha traído muchos cambios en la forma en que son producidos distintos estilos de música, haciendo posible que el artista individual pueda crear su música de una manera práctica y accesible. El MIDI ha logrado esto con bastante éxito al integrar efectivamente el sistema de

producción musical electrónica. A través del uso del MIDI, es posible que cualquier fuente de control (tal como un teclado, almohadilla de tambor, etc) se comuniquen directamente con otro instrumento dentro de un sistema. Usando tal arreglo, es posible comandar más de un instrumento desde un solo controlador (creando de esta manera una textura de sonido más rica y compleja y reduciendo la necesidad de redundantes teclados adicionales).

A través del lenguaje digital de comunicaciones del MIDI, es también posible que los instrumentos y dispositivos musicales electrónicos se comuniquen directamente con una computadora personal. Esto permite al artista utilizar el poder de procesamiento de una computadora personal para ejecutar un amplio rango de tareas de control, edición, grabación y librerías vía MIDI.

## **CAPITULO II INTERFASE DIGITAL PARA INSTRUMENTOS MUSICALES**

La interfase digital para instrumentos musicales es un protocolo digital de comunicaciones. Esto es, un lenguaje de control estandarizado y especificación de hardware que permite a múltiples instrumentos y dispositivos musicales electrónicos comunicar datos de control y ejecución en tiempo real y no real.

### **2.1 ¿Qué es el MIDI?**

El MIDI es un formato de datos específico al que deben adherirse aquellos que diseñan y manufacturan instrumentos equipados con MIDI. De esta manera, los mensajes de automatización y ejecución de tareas pueden comunicarse entre aparatos con relativa transparencia velocidad y facilidad. Así, cuando se ejecuta una tarea (como controlar múltiples instrumentos desde un teclado o transmitir un banco de patch de una librería de patch a un sintetizador), el usuario sólo necesita considerar los parámetros de control de los dispositivos involucrados y aquellos de los mismos medios de transmisión. Esto puede compararse con nuestro lenguaje español, a través del cual se puede transmitir ideas de una persona a una audiencia. Por ejemplo, como personas de habla hispana, somos capaces de concentrarnos totalmente en el contenido de un discurso sin tener que pensar acerca del medio de

lenguaje en sí. Similarmente, los datos de control y performance (entre dispositivos compatibles) son capaces de comunicarse a través del mismo medio de comunicación escogido de MIDI.

### **2.1.1 La palabra digital**

Una de las mejores maneras de visualizar la especificación MIDI es expandir los ejemplos previos de comparar el MIDI con el lenguaje hablado.

Como humanos, hemos adaptado nuestras habilidades de comunicación para adaptarnos a nuestro medio físico. Desde los primeros gruñidos, hemos encontrado más fácil comunicarnos a través del uso de nuestras cuerdas vocales, y así lo hemos estado haciendo desde aquellos días, cuando descubrimos que otra gente o animales podían escuchar nuestros gruñidos y responder a ellos (ojalá positivamente). Así, con el correr del tiempo, se ha desarrollado un lenguaje que asignaba un significado estandarizado a una serie de sonidos (palabras). Eventualmente estas palabras fueron agrupadas para convertirse en un medio de comunicación más complejo. Con el fin de registrar nuestro lenguaje español, se desarrolló un método estandarizado de notación que asignaba 28 símbolos a estos sonidos (letras del alfabeto), los que, cuando se escriben como grupo, comunican una palabra hablada equivalente (Fig 2-1). Al encadenar estas palabras en oraciones completas, podría registrarse una comunicación más compleja. Por ejemplo, las letras A, R, B, O & L no significan nada cuando se

usan individualmente. Sin embargo, cuando se agrupan en una palabra, se refieren a un elemento vegetal conformado por tronco, ramas hojas, etc. Cuando se colocan en una oración, se da a las palabras una mayor claridad de significado (Ej.: "Allá hay un árbol").



**Fig 2-1.** Se da significado a las letras A, R, B, O & L cuando se agrupan en una palabra colocada en una frase.

Por otro lado, los microprocesadores y las computadoras son dispositivos de comunicación digitales que obviamente carecen de cuerdas vocales y oídos (aunque eso está cambiando). Sin embargo, sí tienen la ventaja única de ser capaces de procesar números a gran velocidad. A diferencia de nuestro sistema de contar en base 10, las computadoras están limitadas a comunicarse con un sistema binario de 0s y 1s (o *encendido* y *apagado*). Como los humanos, la mayoría de computadoras son capaces de agrupar estos dígitos (conocidos como *bits*) en valores numéricos más grandes. Estos bits pueden entonces agruparse para formar una palabra digital que

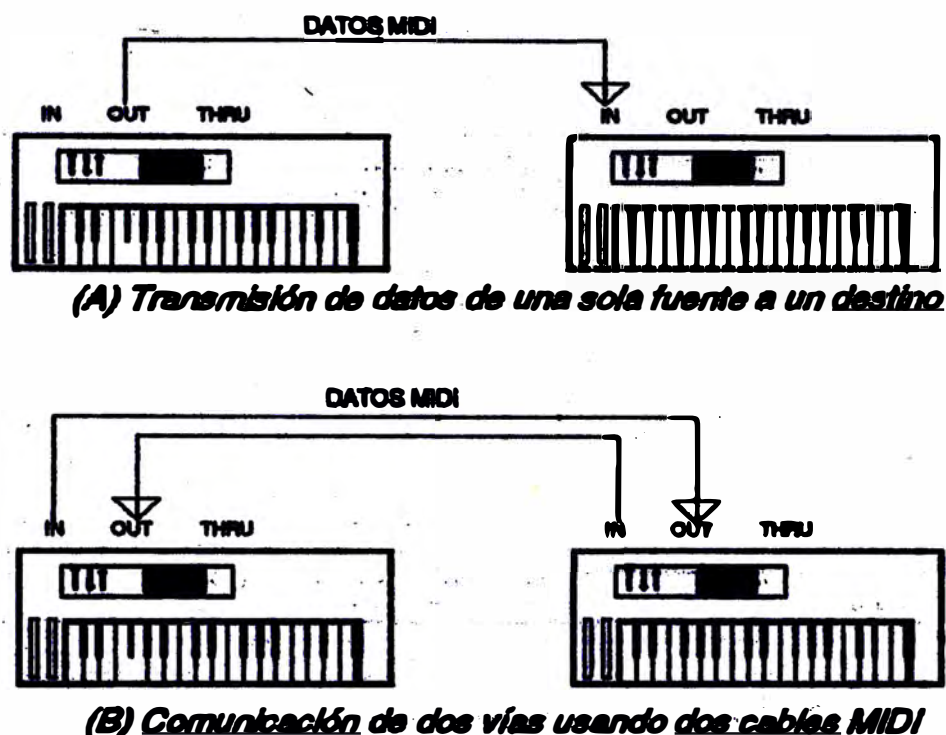
puede usarse para representar y comunicar instrucciones e información específica. Tal como un humano puede comunicar una oración simple, una computadora es capaz de generar y responder a una serie de palabras digitales relacionadas que son entendidas por un sistema digital (Fig. 2-2). La velocidad a la que estos datos son comunicados se mide en *baudios* o número de palabras de 8 bits que se transmiten o reciben por segundo.

**(1001 0100) (0100 0000) (0101 1001)**  
 (BYTE DE ESTATUS)                      (BYTE DE DATOS #1)                      (BYTE DE DATOS #2)

**Fig. 2-2. Ejemplo general de un mensaje MIDI generado digitalmente**

## 2.2 El mensaje MIDI

Los datos de ejecución musical son comunicados digitalmente a través de un sistema de producción como una cadena de mensajes MIDI que se transmite a través de una sola línea MIDI a la velocidad de 31.25 kilobaudios. Los datos de una sola línea MIDI pueden viajar en una sola dirección, de una sola fuente a un destino (Fig. 2-3A). Con el fin de hacer posible una comunicación de dos vías, se debe conectar una segunda línea de datos MIDI desde la fuente esclava de regreso a nuestro original aparato maestro, (Fig. 2-3B).



**Fig. 2-3.** Los datos MIDI sólo pueden viajar en una dirección a través de un solo cable MIDI.

Los mensajes MIDI están hechos de un grupo de palabras de 8 bits (conocidas como *bytes*), que son transmitidas en forma serial para convertirse en una serie de instrucciones a uno o todos los aparatos MIDI dentro del sistema.

Sólo hay dos tipos de bytes que se definen dentro de la especificación MIDI: el *byte de status* y el *byte de datos*. Los bytes de status se usan dentro de un mensaje MIDI como un identificador para instruir al aparato receptor sobre qué función particular y canal MIDI se van a direccionar. El byte de datos se usa para codificar el valor numérico que debe agregarse al byte de status que lo acompaña. Aunque un byte está hecho de 8 bits, el *bit más significativo* (Most significant bit, MSB, el bit más

MSB DE UN BYTE  
DE ESTATUS ES  
SIEMPRE "1"

  
(1SSS SSSS)

MSB DE UN BYTE  
DE DATOS ES  
SIEMPRE "0"

  
(0DDD DDDD)

**Fig. 2-4. El bit más significativo de un byte de datos MIDI se usa para identificar entre un byte de status 1 y un byte de datos 0**

a la izquierda de la palabra digital) se usa únicamente para identificar el tipo de byte. El MSB de un byte de status es siempre 1, mientras que el MSB de un byte de datos es siempre 0 (Fig. 2-4). Por ejemplo, un mensaje de encendido de nota (*note-on*) MIDI de 3 bytes (que se usa para señalar el comienzo de una nota MIDI) en forma binaria se leería:

**TABLA 2.1**

	Byte de status	Byte de datos #1	Byte de datos #2
Descripción	Status/ canal#	Nota #	Velocidad de Ataque
Datos Binarios	(1001 0100)	(0100 0000)	(0101 1001)
Valor Numérico	(Note On/ Canal #4)	(64)	(69)

Así, un mensaje *note-on* de 3 bytes con (10010100) (01000000) (01011001) transmitiría un mensaje que se



leería:

Transmitiendo un mensaje note-on sobre el canal MIDI #4, para la nota #64, con una velocidad de ataque (nivel de volumen de la nota) de 89.

### 2.2.1 Canales MIDI

Tal como es posible que una persona individualice y comunique un mensaje a otra única persona dentro de una multitud, los mensajes MIDI pueden dirigirse a un aparato específico o a un rango de aparatos. Esto se hace al incluir un nibble (cuatro bits) dentro del byte de status/ número de canal (Fig. 2-5) que instruye a todos los aparatos receptores a cuyo canal MIDI se está transmitiendo un mensaje específico. Como el nibble del canal es de 4 bits, a través de un solo cable MIDI se puede transmitir hasta 16 canales.

EL "NIBBLE" FINAL DE 4  
BITS DE UN BYTE DE  
ESTATUS SE USA PARA  
CODIFICAR EL NUMERO  
DE CANAL MIDI



(1SSS CCCC)

**Fig. 2-5.** El nibble menos significativo de un byte de status/ número de canal se usa para codificar el número de canal

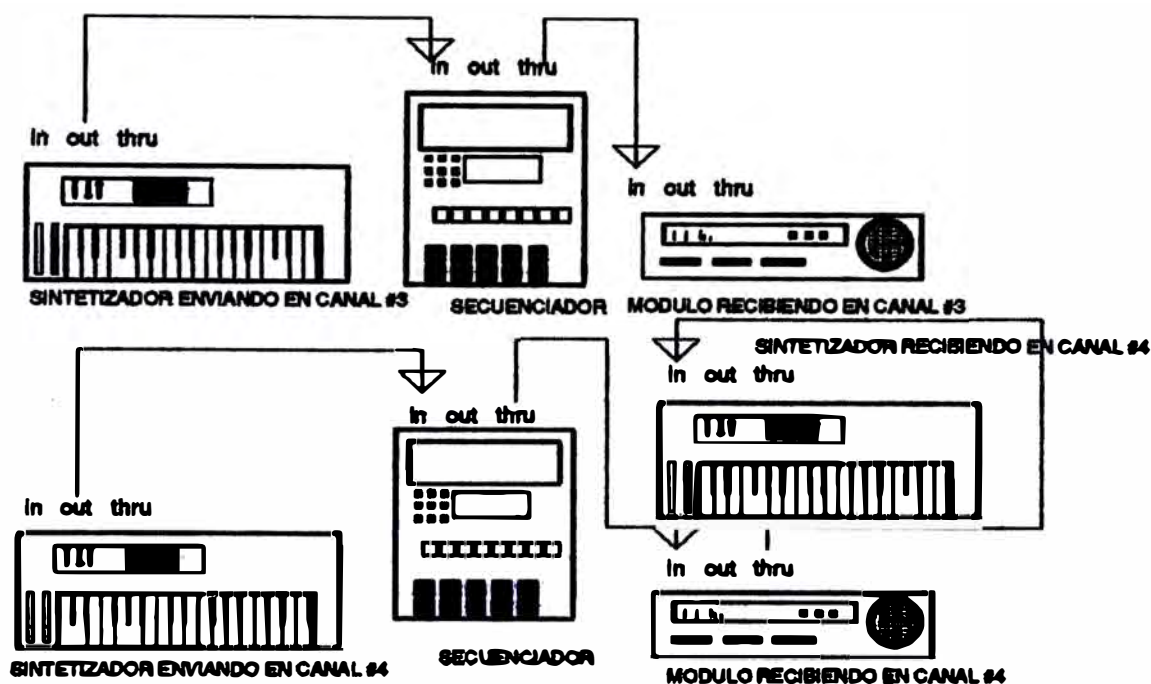
Siempre que a un aparato MIDI se le instruya para responder a un número de canal específico, ignorará los mensajes de canal que sean transmitidos sobre cualquier otro canal. De este modo, cualquier aparato que se

configure para responder a un canal MIDI específico, sólo responderá a mensajes MIDI que sean transmitidos a ese canal (dentro de las capacidades del aparato).

Por ejemplo, asumamos que tenemos dos sintetizadores y un secuenciador MIDI (un aparato que es capaz de grabar, editar y generar una salida de datos MIDI) con los cuales crear una canción corta. Podríamos empezar tocando una línea melódica para nuestro secuenciador en el sintetizador A, que está configurado para transmitir y responder a datos en el canal MIDI #3 (Fig. 2-6). Habiendo hecho esto, podemos decidir tocar acordes de base en el sintetizador B, que se configurará para el canal MIDI #4. Aun cuando el sistema estuviera conectado por una línea MIDI, sería simple que nuestro secuenciador genere salida de los datos MIDI grabados previamente en el canal #3 (que serán tocados todavía en el sintetizador A), mientras nuestro sintetizador B simultáneamente responde a nuestra ejecución en vivo del teclado. Al reproducir la secuencia MIDI terminada, ambos instrumentos responderán sólo a sus canales MIDI asignados, y reproducirán sus sonidos individuales tal como fueron grabados.

### **2.2.2 Modos MIDI**

Los instrumentos electrónicos frecuentemente varían en el número de sonidos que pueden producirse a un tiempo usando su circuitería interna de generación de sonidos. También pueden variar en el número de sonidos característicos individuales que pueden ser



**Fig. 2-6.** Sistema que muestra un juego de asignaciones de canal MIDI.

simultáneamente producidos por un instrumento. Por ejemplo, ciertos instrumentos son capaces de producir solamente una nota a la vez, mientras otros (conocidos como *instrumentos polifónicos*) son capaces de generar numerosas notas a un tiempo. Esto último permite al artista tocar acordes y más de una línea musical en un solo instrumento. Adicionalmente, es posible que varios tipos de sintetizadores produzcan un solo patch de sonido característico a un tiempo (ej., piano eléctrico, bajo sintetizado, etc.). La palabra *patch* (clavijero, traducido del inglés), es una alusión directa a la necesidad de *patch chords*, cuando se usaban los antiguos sintetizadores analógicos, para conectar un generador de sonido o procesador con otro. Por otro lado, un solo

instrumento puede ser también *multitimbral* por naturaleza, lo que significa que es capaz de generar más de un patch de sonido a la vez.

Como resultado de estas diferencias entre aparatos, se ha especificado una serie de lineamientos definidos (conocidos como *Modos de Recepción MIDI*) que permite que un instrumento transmita o responda a mensajes de canal MIDI de varias maneras. Por ejemplo, un instrumento puede estar programado para responder a todos los 16 canales MIDI al mismo tiempo, mientras otro puede ser polifónico por naturaleza y estar programado para responder a sólo un canal MIDI. También es común que un instrumento sea polifónico y multitimbral (lo que permite que cierto número de patches de sonido generados respondan individualmente a sus propios canales MIDI asignados). Hacemos un listado de estos modos:

Modo 1 - Omni on/poly

Modo 2 - Omni on/mono

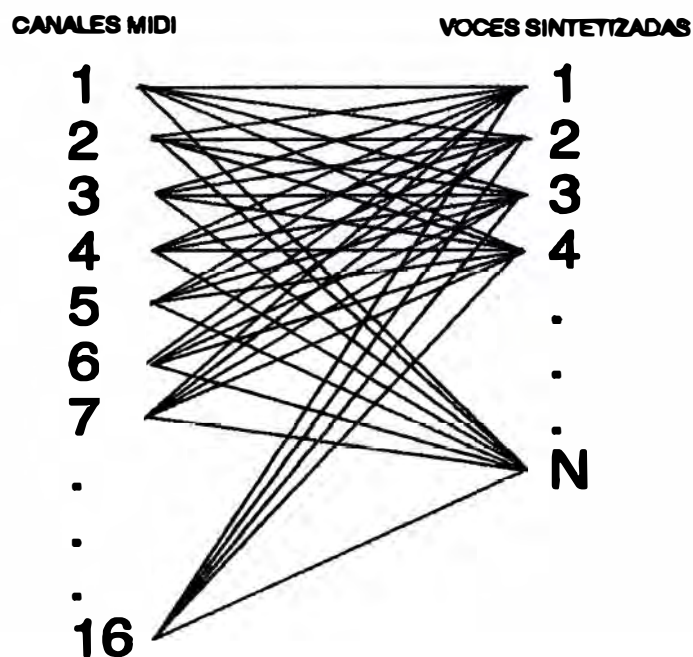
Modo 3 - Omni off/poly

Modo 4 - Omni off/mono

*Omni on/off* se refiere a cómo un instrumento MIDI responderá a los 16 canales MIDI. Cuando Omni se enciende (*on*), el aparato MIDI responderá a todos los mensajes de

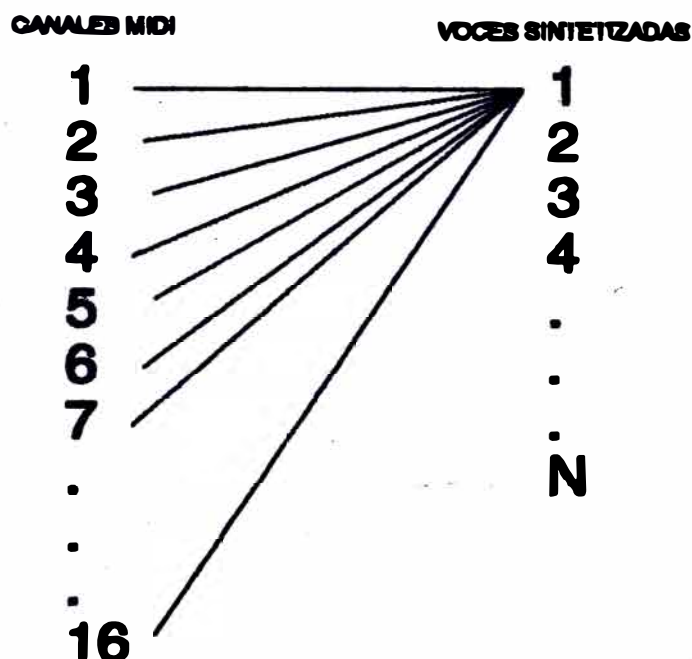
canal que sean transmitidos sobre todos los canales MIDI. Siempre que Omni se apaga (*off*), el aparato sólo responderá a un solo canal MIDI o juego de canales asignados. *Poly/mono* se refiere a la generación de notas individuales por un instrumento MIDI. En el modo *Poly*, un instrumento es capaz de responder polifónicamente a cada canal MIDI y es capaz de producir más de una nota a la vez. En el modo *Mono*, un instrumento es capaz de responder monofónicamente a cada canal MIDI y es capaz de producir sólo una nota a la vez.

*\*Modo 1 - Omni on/poly:* Un instrumento será capaz de responder polifónicamente a los datos de ejecución que se recibe en cualquier canal MIDI (Fig. 2-7).



**Fig. 2-7.** Ejemplo de asignación voz/canal del modo 1 - Omni on/poly

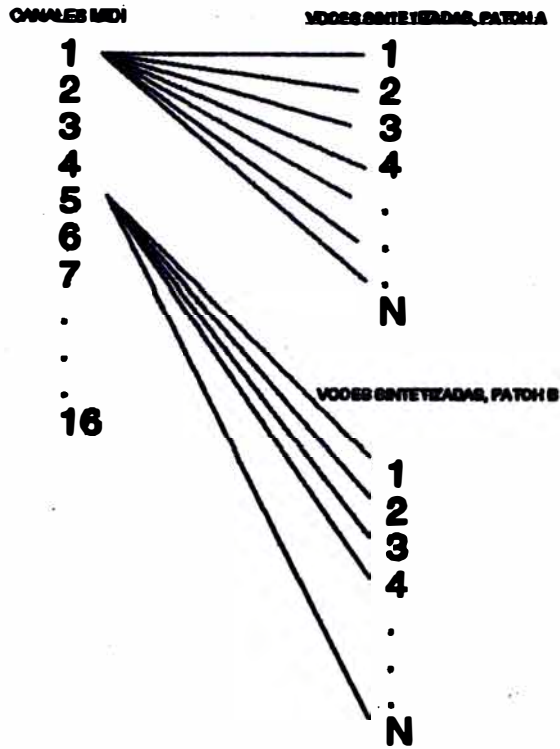
**\*Modo 2 Omni on/mono:** Un instrumento asignará cualquier evento de nota recibida a una voz monofónica, sin considerar en qué canal MIDI es recibido (Fig. 2-8). Este modo raramente se usa.



**Fig. 2-8. Ejemplo de asignación voz/canal del modo 2 - Omni on/mono**

**\*Modo 3 - Omni off/poly:** Un instrumento será capaz de responder polifónicamente a datos de ejecución que se transmiten sobre uno o más canales asignados (Fig. 2-9). Como tal, este modo es comúnmente usado por instrumentos politimbrales.

**\*Modo 4 - Omni off/mono:** Un instrumento sólo será capaz de generar una nota MIDI por canal; (Fig. 2-10). Un ejemplo práctico de este modo se usa frecuentemente



**Fig. 2-9. Ejemplo de asignación de canal/voz del modo 3 - Omni off/poly**



**Fig. 2-10. Ejemplo de asignación voz/canal del modo 4 - Omni off/mono.**

en sistemas de guitarra MIDI, como sea que los datos MIDI son transmitidos monofónicamente sobre seis canales consecutivos (una voz/canal por cuerda). Otros instrumentos electrónicos pueden hacer uso de este modo y permiten que voces individuales sean asignadas a cualquier combinación de canales MIDI.

### **2.2.3 Mensajes canal-voz**

*Los mensajes canal-voz* se usan para transmitir datos de ejecución en tiempo real a través de un sistema MIDI conectado. Se generan cada vez que el controlador de un instrumento MIDI es tocado, seleccionado o variado por el ejecutante. Ejemplos de tales cambios de control serían el tocar un teclado, botones de selección de programa, o movimiento de las ruedas de modulación o pitch. Cada mensaje canal-voz contiene un número de canal MIDI dentro de su byte de status y es capaz de generar respuesta de un aparato que esté asignado al mismo número de canal. Hay siete tipos de mensajes canal-voz: note-on, note-off, presión de teclado polifónico, presión de canal, cambio de programa, cambio de control y cambio de curva de pitch (pitch-bend).

#### **Note On**

Un mensaje note-on se usa para indicar el comienzo de una nota MIDI. Se genera cada vez que una nota es disparada en un teclado, batería electrónica u otro instrumento MIDI (al presionar una tecla, almohadilla percusiva, o al tocar una secuencia).

Un mensaje note-on consta de tres bytes de información



(Fig. 2-11): un byte de status/número de canal MIDI de note-on, número de pitch MIDI y valor de velocidad de ataque.



**Fig. 2-11. Estructura de bytes de un mensaje MIDI note-on.**

El primer byte dentro del mensaje especifica un evento de note-on y un canal MIDI (1-16). El segundo byte se usa para especificar cual de los posibles 128 pitches (numerados de 0-127) se hará sonar en el instrumento MIDI. En general, el número de nota 60 se asigna a la tecla DO central de un teclado igualmente temperado, mientras las notas 21-108 corresponden a las 88 teclas de un controlador de teclado extendido.

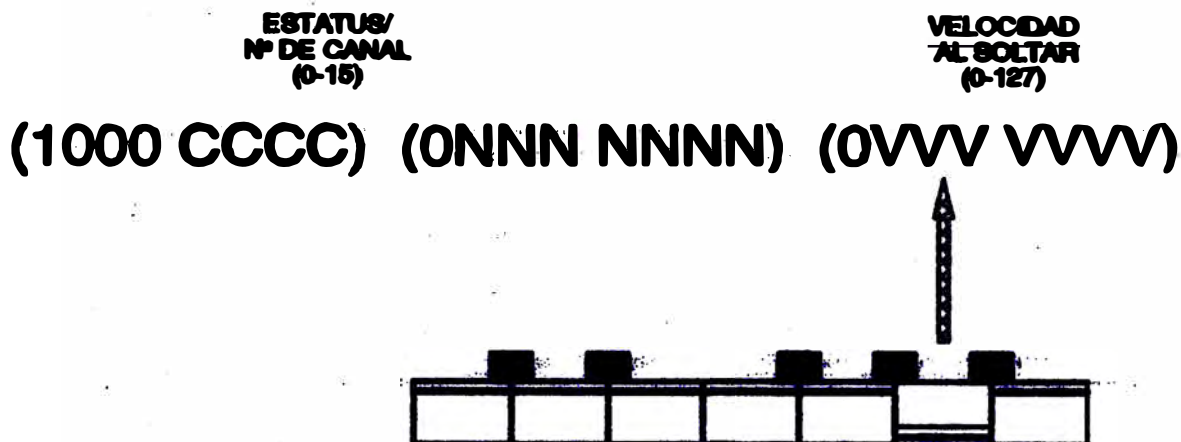
El byte final se usa para indicar la rapidez o velocidad a la que la tecla fue presionada. El rango de este valor comúnmente varía de 1-127. Además se usa para

denotar el volumen de la nota generada, que se incrementa en volumen con mayores valores de velocidad. No todos los instrumentos se diseñan para interpretar el rango completo de valores de velocidad (como ciertas baterías electrónicas), mientras otros no responden dinámicamente para nada. Los instrumentos que no contienen información de velocidad, generalmente transmitirán un valor de velocidad de ataque de 64 para cada nota que es tocada, sin considerar la velocidad real. Similarmente, instrumentos que no responden a mensajes de velocidad interpretarán todos los mensajes de velocidad MIDI como si tuvieran un valor de 64.

Un mensaje *note-on* que contiene una velocidad de ataque de 0 (cero), es generalmente equivalente a la transmisión de un mensaje *note-off*. Esto indica que el aparato va a silenciar una nota actualmente sonando al tocarla con un nivel de velocidad (volumen) de 0.

### **Note Off**

Un mensaje *note-off* se usa como un comando para detener la ejecución de una nota específica MIDI. Cada nota que ha sido tocada vía un mensaje *note-on*, se sostiene hasta que se recibe el correspondiente mensaje *note-off*. De este modo, una ejecución musical puede ser codificada como una serie de mensajes MIDI *note-on* y *note-off*. También debe tenerse en cuenta que un mensaje *note-off* no cortará el sonido, solamente parará de tocarlo. Si el patch que se está tocando tiene una etapa de *release* (o caída final), esta empezará al recibir este



**Fig. 2-12. Estructura de bytes de un mensaje MIDI note-off.**

mensaje.

Al igual que el mensaje note-on, la estructura del note-off consta de tres bytes de información (Fig. 2-12): un byte de status/número de canal MIDI de note-off, número de pitch MIDI (que corresponden a los mismos números que son direccionados por el mensaje note-on) y valor de velocidad de release (o velocidad al soltar).

Análogamente a la dinámica de la velocidad de ataque, el valor de velocidad al soltar (*release* 0-127) indica la rapidez o velocidad a la que la tecla fue soltada. Un valor bajo indica que la tecla fue soltada muy lentamente, mientras que un valor alto muestra que la tecla fue soltada rápidamente. Aunque pocos instrumentos generan o responden a mensajes de velocidad al soltar MIDI, los que son capaces de responder a estos valores pueden programarse para que varíe la velocidad de caída

(*decay*) de una nota, con frecuencia reduciendo el tiempo de caída de la señal cuando se incrementa el valor de la velocidad al soltar.

### **Presión de tecla polifónica**

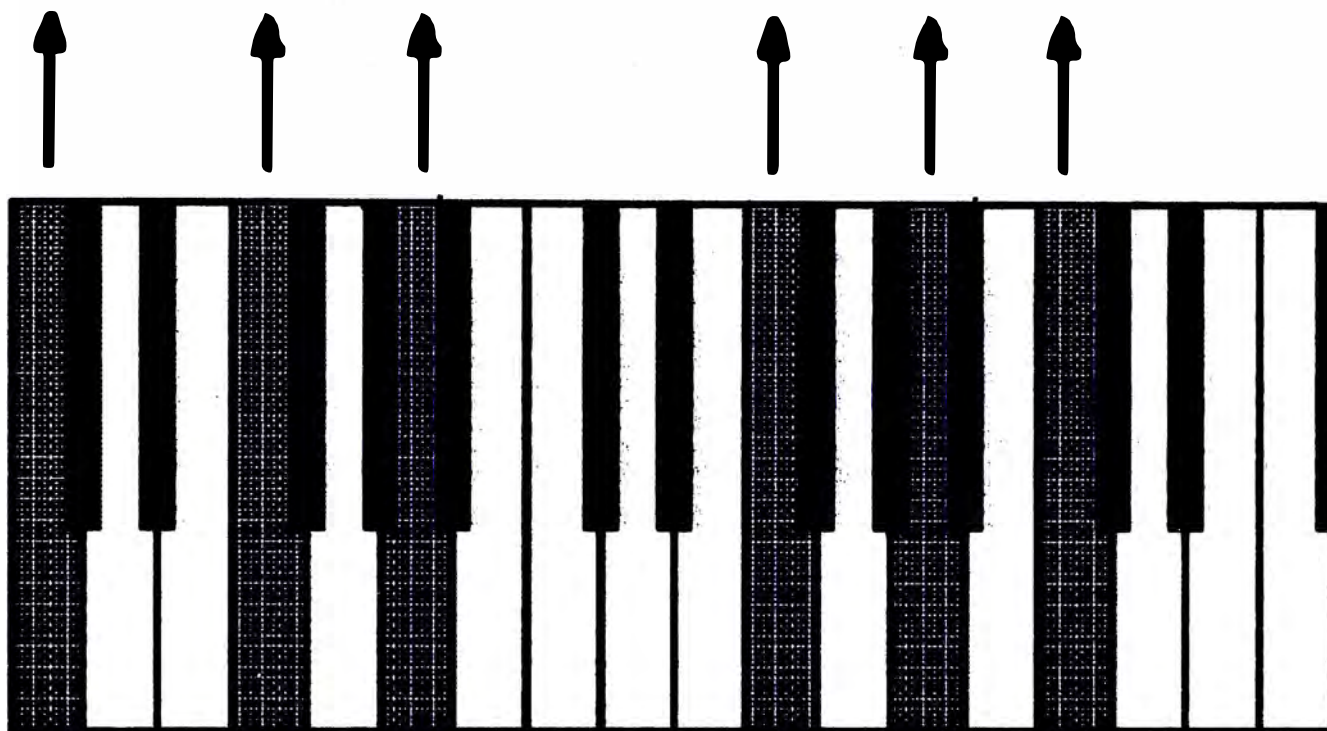
Los *mensajes de presión de tecla polifónica* son transmitidos comúnmente por instrumentos capaces de responder a los cambios de presión que se aplican a las teclas individuales de un teclado. Tal instrumento puede usarse para transmitir mensajes de presión individual para cada tecla que se oprime. (Fig. 2-13).

Un mensaje de presión de tecla polifónica consta de tres bytes de información (Fig. 2-14): un byte de status/número de canal MIDI de presión de tecla polifónica, uno de número de pitch MIDI y un valor de presión.

El medio por el cual un aparato responde a estos mensajes varía frecuentemente entre diferentes fabricantes. Sin embargo, los valores de presión pueden comúnmente asignarse a parámetros de ejecución tales como vibrato, volumen, timbre y pitch. Aunque los controladores que son capaces de producir presión polifónica son pocos y generalmente más caros, no es inusual que un instrumento responda a estos mensajes.

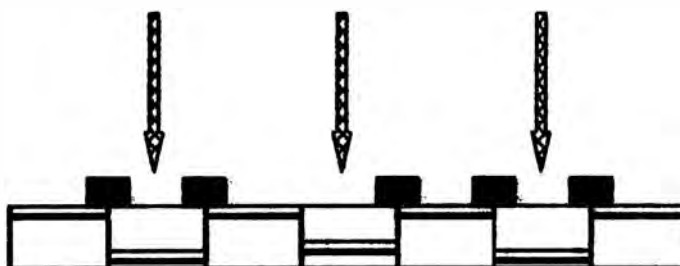
### **Presión de Canal (After Touch)**

Los *mensajes de presión de canal* (generalmente conocidos como *after touch*, después del toque) son comúnmente transmitidos por instrumentos que sólo responden a una presión simple y completa aplicada a sus



**Fig. 2-13.** Los mensajes de presión de tecla polifónica individual son generados cuando se aplica presión adicional a cada tecla tocada.

ESTATUS/ Nº DE CANAL (0-15)	VALOR DE PRESION (0-127)	
(1010 CCCC)	(0NNN NNNN)	(0VVV VVVV)



**Fig. 2-14.** Estructura de bytes de un mensaje de presión de tecla polifónica MIDI.

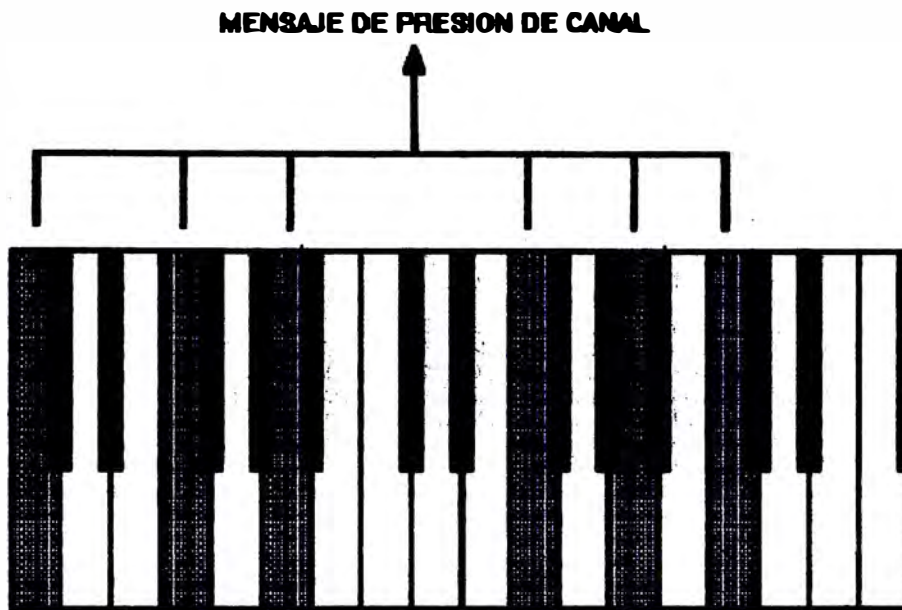
controladores, sin considerar el número de teclas que se toca a la vez (Fig 2-15). Por ejemplo, si se toca seis notas en un controlador de teclado y se aplica presión adicional a sólo una tecla, las seis notas se verán afectadas. Un mensaje de presión de canal consta de tres bytes de información (Fig. 2-16): un byte de status/número de canal MIDI de presión de canal, número de pitch MIDI y valor de presión.

Tal como en los cambios de presión polifónica, un instrumento puede frecuentemente programarse para responder a mensajes de presión de canal en muchas formas. Por ejemplo, los valores de presión de canal pueden comúnmente asignarse a parámetros de ejecución tales como vibrato, volumen, corte de filtro y pitch.

#### **Cambio de programa**

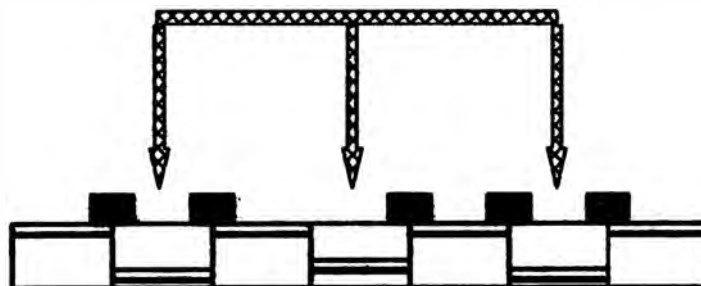
El *mensaje de cambio de programa* se usa para cambiar el programa o número de pre-configuración (preset number) que se encuentra activo en un aparato o instrumento MIDI. Un *preset* es un número definido por el usuario o fábrica que activa un patch de sonido específico o configuración de sistema. Se puede seleccionar hasta 128 presets vía MIDI usando este mensaje. Un mensaje de cambio de programa (Fig. 2-17) consta de dos bytes de información: un byte de status/número de canal de cambio de programa (1-16) y un número identificador ID de programa (0-127).

Como ejemplo, un mensaje de cambio de programa puede usarse para conmutar entre los varios patches de sonido de un sintetizador desde un instrumento de teclado remoto



**Fig. 2-15.** Un mensaje de presión de canal afecta a todas las notas sostenidas que se transmiten sobre cada canal MIDI.

<b>ESTATUS/ Nº DE CANAL (0-15)</b>	<b>VALOR DE PRESION (0-127)</b>
<b>(1010 CCCC)</b>	<b>(0NNN NNNN) (0VVV VVVV)</b>

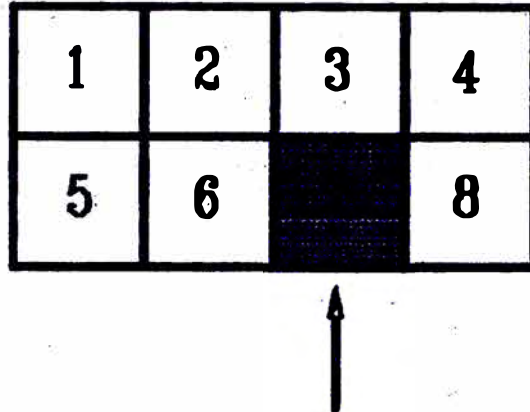


**Fig. 2-16.** Estructura de bytes de un mensaje de presión de canal MIDI.

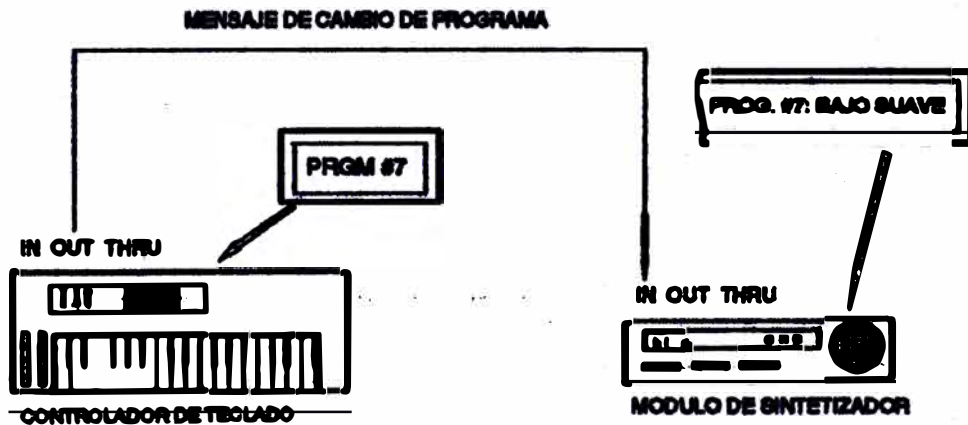


ESTATUS/  
Nº DE CANAL  
(0-16)

(1100 NNNN) (0PPP PPPP)



**Fig. 2-17.** Estructura de bytes de un mensaje de cambio de programa MIDI.



**Fig. 2-18.** Los mensajes de cambio de programa se usan para cambiar patches de sonido desde un controlador remoto.



(Fig. 2-18). También puede usarse para seleccionar patrones rítmicos y/o configuraciones dentro de una batería electrónica, llamar patches específicos de efectos dentro de un aparato de efectos, o una multitud de otras configuraciones del controlador y el sistema que puede llamarse como preset de programa. Muchos aparatos MIDI permiten activarse y desactivarse manualmente por el usuario para reconocer estos mensajes.

### **Cambio de Control**

El *mensaje de cambio de control* se usa para transmitir información que se relaciona al control en tiempo real sobre los parámetros de ejecución de un instrumento MIDI. Hay tres tipos de controles en tiempo real que pueden comunicarse vía mensajes de cambio de control:

*-Controladores continuos.-* Controladores que manejan un rango completo de posiciones de control. Comúnmente van de 0 a 127. Sin embargo, dos mensajes de controlador pueden combinarse en tándem para lograr una resolución mayor.

*-Switches (conmutadores):* Controladores que se encuentran en estado de encendido (on) o apagado (off) sin posiciones intermedias.

*-Controladores de datos:* Controladores que ingresan datos ya sea a través del uso de un teclado numérico o aumentando y disminuyendo a través del uso de botones de

ingreso de datos.

Un mensaje de cambio de control simple, o una sucesión de tales mensajes, se transmite cada vez que los controladores (tales como pedales de switch, ruedas de cambio de afinación, ruedas de modulación, controladores de respiración, etc.) son variados en tiempo real. De este modo, un controlador puede usarse para variar correspondientemente un amplio rango de parámetros posibles dentro de un instrumento o dispositivo de acuerdo con los movimientos o comandos del controlador. Un mensaje de cambio de control (Fig 2-19) consta de tres bytes de información: un byte de status/número de canal de cambio de control (1-16), una número ID de controlador (0-127) y el valor de controlador correspondiente (0-127).

STATUS/ NUMERO DE CANAL	NUMERO DE ID DEL CONTROLADOR	VALOR DEL CONTROLADOR
<b>(1011 NNNN)</b>	<b>(0CCC CCCC)</b>	<b>(0VVV VVVV)</b>

**Fig. 2-19. Estructura de bytes de un mensaje de cambio de control MIDI**

***Número ID del controlador***

El segundo byte de el mensaje de cambio de control se usa para denotar el número de identificación del

controlador. Este número se usa para especificar cual de los parámetros de programa o performance del dispositivo se va a direccionar.

Aunque muchos fabricantes siguen una convención general para asignar números de controlador a parámetros asociados, están libres de asignar estos parámetros como deseen, siempre que sigan el formato definido propuesto por la especificación MIDI (Fig. 2-20).

### ***Valores de controlador***

El tercer byte del mensaje de cambio de control se usa para denotar los valores instantáneos del controlador. Este valor se usa para especificar la posición, profundidad, o nivel del efecto que el controlador tendrá sobre el parámetro. En la mayoría de los casos, el rango de valores de un controlador continuo de 7 bits caerá entre 0 (mínimo) y 127 (máximo valor) (Fig. 2-21). El rango de valores de un controlador de switch es frecuentemente 0 (off) y 127 (on) (Fig. 2-22A). Sin embargo, las funciones de switch también son capaces de responder a mensajes de controladores continuos al reconocer valores de 0-63 como off, y de 64-127 como on (Fig. 2-22B).

La práctica de usar valores de 0-127 para representar un incremento de profundidad de efecto o nivel de señal no pertenece a los parámetros de control de balance, panorama y expresión.

Un *controlador de balance* se usa para variar los niveles relativos entre dos fuentes de sonido

Faders (atenuadores) de datos

Rueda de datos

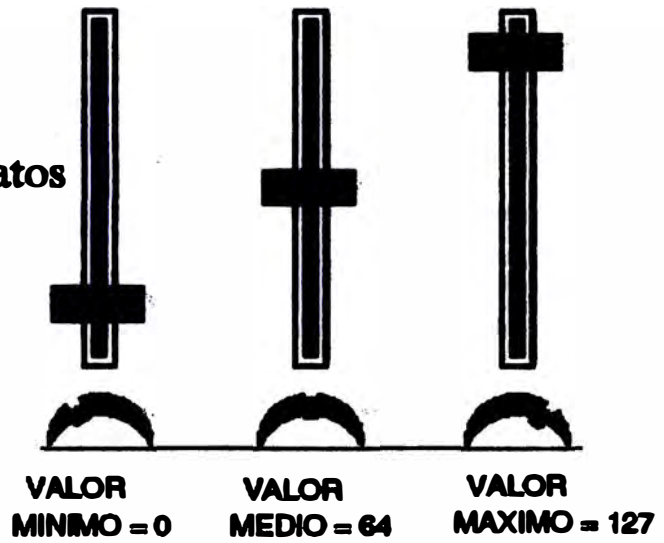
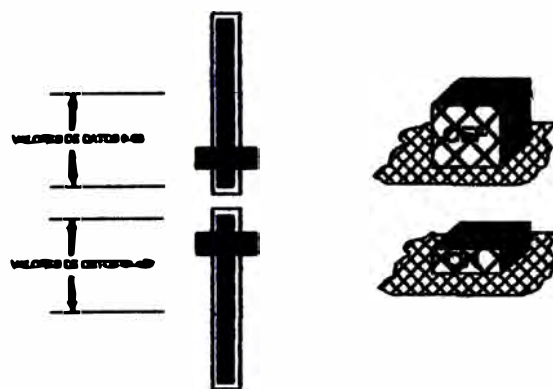
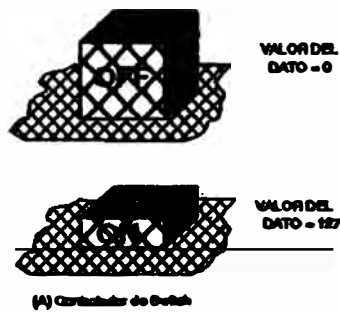
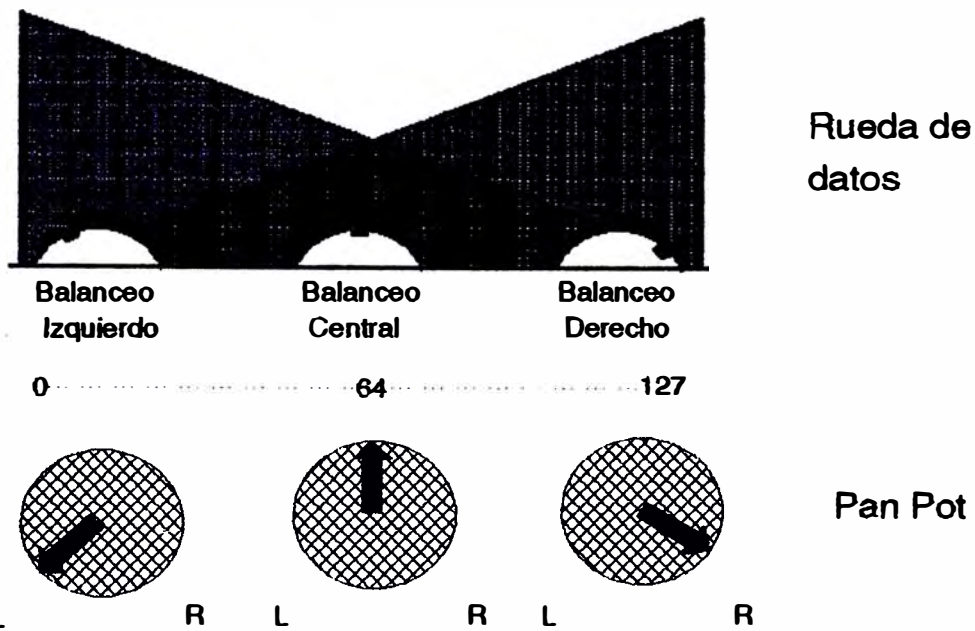


Fig. 2-21. Rangos de valores del controlador continuo.



(B) Un parámetro de switch respondiendo a mensajes de controlador continuo

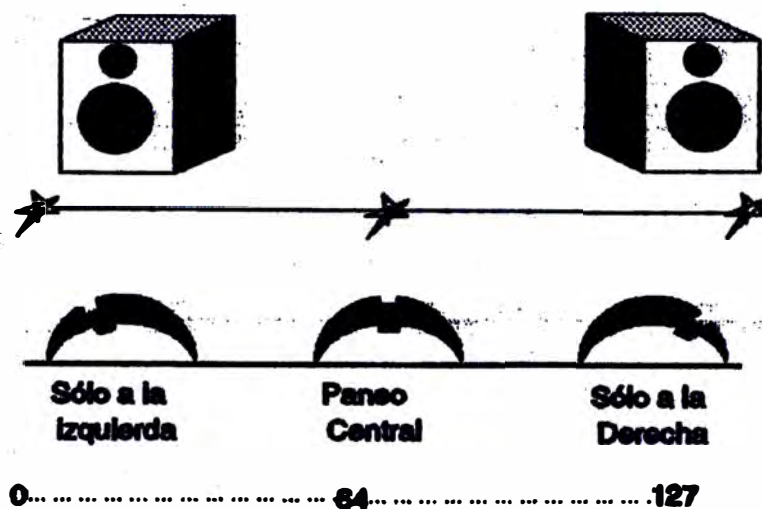
Fig. 2-22. Rangos de valores de datos del controlador de switch



**Fig. 2-23.** Rango de valores de datos del controlador de balance.

independientes (Fig. 2-23). Así como en el control de balance de un amplificador estéreo, este controlador se usa para ajustar el balance relativo izquierda/derecha de una fuente de sonido estéreo. El rango de valores de este controlador cae entre 0 (todo a la fuente de sonido izquierda) y 127 (todo a la fuente de sonido derecha). Con un valor de 64 para representar un campo balanceado estéreo izquierda/derecha.

Un *controlador de panorama* (pan controller) se usa para posicionar el balance relativo de una fuente simple de sonido entre los canales izquierdo y derecho de un campo de sonido estéreo (Fig. 2-24). El rango de valores de



**Fig. 2-24. Rango de valores de datos del control de panorama.**

este controlador cae entre 0 (posición a la izquierda) y 127 (posición a la derecha), con un valor de 64 para representar un campo central balanceado.

Un *controlador de expresión* se usa para acentuar las posiciones de nivel existentes en un instrumento o dispositivo MIDI. Este control puede usarse para incrementar el nivel de volumen de canal de un instrumento, pero no puede reducir este nivel por debajo de su posición de volumen programada. El rango de valores de este controlador cae entre 0 (actual posición de volumen programada) y 127 (acento de volumen total).

#### ***Formato del ID de controlador***

Los siguientes formatos de ID de controlador han sido definidos por la especificación MIDI 1.0 para asegurar compatibilidad entre fabricantes con respecto a los variados parámetros de control y sus mensajes.

*Controladores de 14 bits:* Los números de controlador 0-31 se reservan comúnmente para mensajes estándares de controlador continuo (tales como modulación, respiración, volumen principal, panorama, etc.). Sin embargo, si se requiriera una gran resolución, es posible ligar en forma correspondiente estos mensajes con números de mensaje 32-63. Así, en lugar de transmitir dos bytes de datos sobre un mensaje, se transmitirá cuatro bytes sobre dos mensajes. Esto eleva la resolución de un total de 127 pasos a 16883 pasos.

*Controladores de 7 bits:* Los números de controlador 64-95 se reservan comúnmente para funciones de switch (tales como pedal de sordina, portamento y sustain), o aquellas relacionadas a la profundidad de un efecto programado (tal como trémolo, chorus y phase). Como los mensajes adicionales no están reservados para estas funciones, su rango de valores está limitado a 0-127.

*Controladores de propósito general:* Estos mensajes no están asignados a ninguna función de controlador particular y pueden ser asignados a cualquier controlador de un dispositivo específico a discreción del fabricante.

*Controladores indefinidos:* Los mensajes indefinidos no son asignados actualmente a ninguna función de controlador particular. Sin embargo, están reservados para futuros parámetros de controlador.

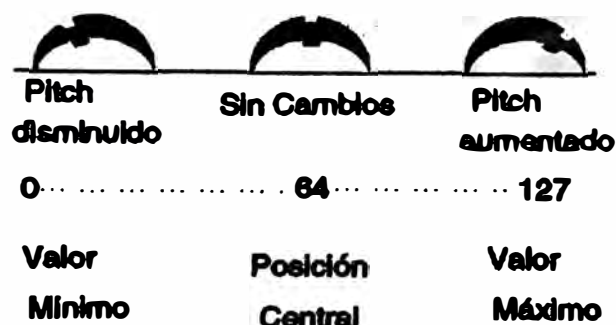
*Controladores de incremento de datos:* Estos mensajes se usan para transmitir incremento (+) o decremento (-) de datos en los controles afines.

*Controles registrados y no registrados:* Sólo tres tipos de controladores registrados y no registrados son asignados actualmente a una función de controlador. Estos son: sensibilidad de curva de pitch (pitch-bend), afinación fina (fine tuning) y afinación gruesa (coarse tuning).

*Sensibilidad de curva de pitch (pitch-bend)* se refiere a la sensibilidad de respuesta (en semitonos) de una rueda de curva de pitch (pitch-bend wheel). El rango de valores de 7 bits de este controlador permite ajustes en incrementos de hasta  $1/128$  del rango de curva de pitch. La afinación fina permite ajustes controlados (sea hacia arriba o abajo) al dividir un semitono en 8192 pasos posibles de afinación o altura tonal. La afinación gruesa provee un rango de afinación máximo de hasta 63 semitonos arriba y 64 semitonos abajo del valor de afinación estándar.



### Cambio de Curva de pitch (pitch-bend)



**Fig. 2-25.** Rango de valores de datos de la rueda de afinación.



**Fig. 2-26.** Estructura de bytes de un mensaje de pitch bend.

Los *mensajes de cambio de Curva de pitch (pitch-bend)* son transmitidos por un instrumento cada vez que su rueda de curva de pitch (Fig. 2-25) se mueva hacia la posición positiva (elevar pitch) o negativa (disminuir pitch) desde su posición central (no hay cambio de pitch). Los mensajes de curva de pitch consisten en tres bytes de información (Fig. 2-26): un número de canal MIDI, un bit más significativo (MSB) y un bit menos significativo

(LSB). Dado que estos datos se transmiten sobre dos bytes de datos, este mensaje tiene una resolución total de 14 bits.

#### **2.2.4 Mensajes canal-modo**

Los números de controlador 121-127 están reservados para *mensajes de canal modo*. Estos incluyen controladores de reset completo, control local, todas las notas off, y mensajes de modo MIDI.

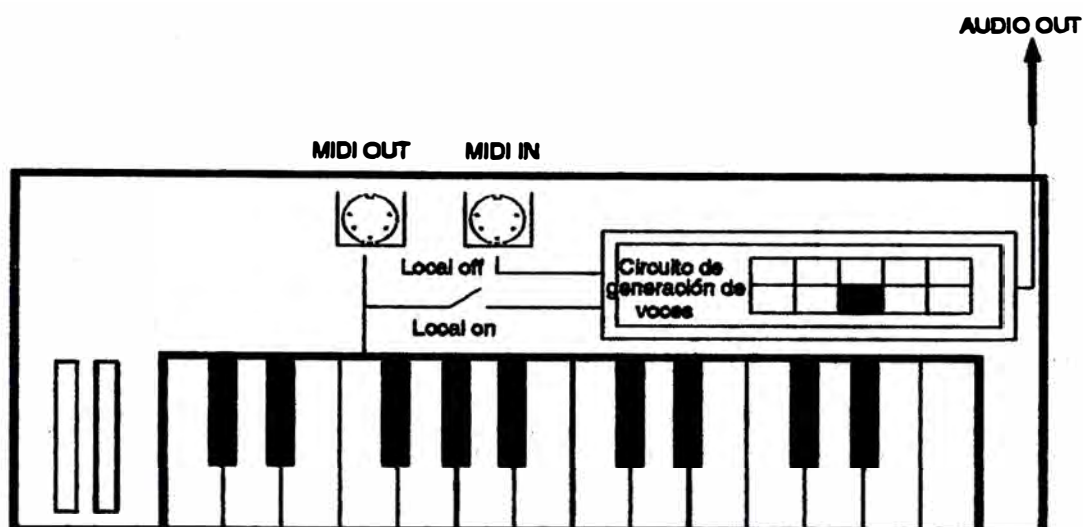
##### **Controladores de reset completo**

Este mensaje se usa para reinicializar todos los controladores (continuos, switch, e incrementales) dentro de uno o más instrumentos o dispositivos receptores MIDI a un estado de encendido por defecto estándar.

##### **Control local**

Los *mensajes de control local* se usan para desconectar el controlador de un instrumento MIDI de sus propias voces (Fig. 2-27). Esta característica es útil para hacer que un teclado sea el control de performance maestro de un sistema MIDI. Aunque la circuitería de sonido de un instrumento pueda ser desconectada de su controlador interno (cuando el control local de un instrumento se apaga), este es aun capaz de responder a mensajes de control y performance MIDI que vienen desde un controlador externo o secuenciador.

Un mensaje de control local consta de dos bytes de información: un número de canal MIDI (1-16), y un byte de status de control local on\off.



**Fig. 2-27. Función Encendido/Apagado de control local**

### Todas las notas off

Ocasionalmente, un mensaje de note-on será recibido por un instrumento MIDI, y el consiguiente mensaje de note-off es de algún modo ignorado o no recibido. Este infortunado evento frecuentemente da como resultado una nota pegada que continuará sonando hasta que se reciba un mensaje de note-off para ese pitch. Como una alternativa a estar buscando esa nota, se puede transmitir un mensaje de *todas las notas off* (all notes off), que efectivamente apaga todas las 128 notas. Como no todos los instrumentos MIDI son capaces de responder a este mensaje del botón de pánico, debería transmitirse como último recurso.

### Modo Omni off

Al recibir un mensaje de *modo Omni off* (Omni mide off) un instrumento o dispositivo MIDI cambiará de modo (o permanecerá en el modo Omni-off), para que responda a

canales MIDI asignados individualmente en lugar de responder a todos los canales MIDI a la vez.

### **Modo Omni on**

Al recibir un mensaje de *modo Omni on* (Omni mode on) un instrumento o dispositivo MIDI cambiará de modo (o permanecerá en el modo Omni-on), para que respondá a todos los mensajes de canal MIDI, sin considerar a qué canales están siendo transmitidos estos mensajes.

### **Modo Mono on**

Al recibir un mensaje de *modo mono on*, un instrumento MIDI asignará voces individuales a canales MIDI consecutivos, empezando por el más bajo actualmente asignado o canal básico. Esto es, el instrumento puede tocar sólo una nota por cada canal MIDI. Sin embargo, es capaz de tocar más de un canal monofónico al mismo tiempo.

### **Modo Poly on**

Al recibir un mensaje de *modo poly on* (poly mide on) un instrumento o dispositivo MIDI cambiará de modo (o permanecerá en el modo poly on). Este mensaje permite que un instrumento responda polifónicamente a los canales MIDI. De este modo, un dispositivo es capaz de tocar más de una nota a la vez sobre un canal o número de canales dado.

## **2.2.5 Mensajes de sistema**

Como su nombre lo indica, los *mensajes de sistema* se transmiten globalmente a todos los dispositivos MIDI de la cadena MIDI. Esto se logra porque los números de canal

MIDI no se direccionan dentro de la estructura de bytes de un mensaje de sistema. Este hecho significa que cualquier dispositivo responderá a estos mensajes, sin considerar a cuál canal o canales MIDI sean asignados. Existen tres tipos de mensajes de sistema: mensajes comunes de sistema, mensajes de tiempo real de sistema y mensajes exclusivos de sistema.

#### **Mensajes comunes de sistema.**

Los *mensajes comunes de sistema* (system-common messages) se usan para transmitir código de tiempo MIDI, cursor de posición de canción, selector de canción, pedido de afinación y datos de final de exclusivo a través de todo el sistema MIDI o los 16 canales de un puerto MIDI especificado.

#### ***MTC Quarter Frame***

El *código de tiempo MIDI* (MTC) provee un medio barato y fácilmente implementable para traducir el código SMPTE (el código de la *Society of Motion Picture and Television Engineers* o *Standardized Synchronization Time Code*) a un código equivalente que se adecua a la especificación MIDI 1.0. Permite que los comandos y códigos basados en el tiempo se distribuyan a través de toda la cadena MIDI. Los mensajes de *MTC Quarter Frame* son transmitidos y reconocidos por dispositivos MIDI que sean capaces de entender y ejecutar comandos MTC.

Se requiere un grupo de ocho Quarter Frames para denotar una dirección de código de tiempo completa (en horas, minutos, segundos y Frames). Por esta razón, el

tiempo SMPTE se actualiza cada dos Frames. Cada mensaje de Quarter Frame contiene dos bytes.

El primero es el encabezado común de un Quarter Frame. El segundo byte contiene un nibble (cuatro bits) que representa el número de mensaje (0-7) y un nibble final para cada uno de los dígitos de un campo de tiempo (horas, minutos, segundos o Frames)

### ***Cursor de posición de canción***

El *cursor de posición de canción* (song position pointer SPP) permite que un secuenciador o máquina de ritmos pueda ser sincronizada a una fuente externa (tal como una grabadora de cinta) desde cualquier posición dentro de una canción. El mensaje SPP se usa para lograr la referencia de un punto de locación sobre un dispositivo externo (tales como una grabadora de cinta o una máquina de ritmos). Este mensaje provee una referencia de tiempo que se incrementa una vez por cada seis mensajes de reloj MIDI, con respecto al comienzo de la composición.

Un mensaje SPP se transmite generalmente mientras la secuencia MIDI se ha detenido, permitiendo que los dispositivos MIDI equipados con SPP persigan con gran velocidad la canción y se enganchen a la fuente externa una vez que se logra el sincronismo relativo.

### ***Selector de canción***

Un mensaje de *selector de canción* se usa para buscar una canción específica en la memoria de secuencia interna de una máquina de ritmos o secuenciador (al identificarla

por su número ID de canción). Al ser seleccionada, la canción responderá a los mensajes de empezar, detenerse y continuar.

### ***Pedido de afinación***

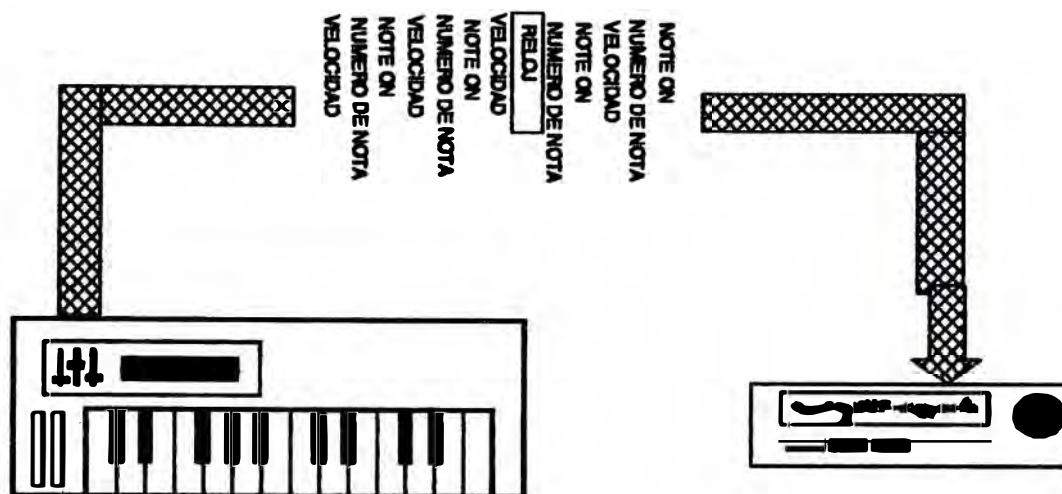
Este mensaje se usa para pedir a un instrumento MIDI que inicie su rutina interna de afinación (si está equipada para esto).

### ***Final de Exclusivo***

La transmisión de un mensaje de *final de exclusivo* (end-of-exclusive EOX) se usa para indicar el final de un mensaje exclusivo de sistema. Le daremos mayor cobertura a este tema al final de este capítulo.

### **Mensajes de tiempo real de sistema**

La transmisión de mensajes de tiempo real de sistema provee el elemento de tiempo preciso que los dispositivos MIDI requieren para su sincronización durante la performance. Para evitar la introducción de retrasos en el tiempo, la especificación MIDI permite que los mensajes de tiempo real sean insertados en cualquier punto dentro del flujo de datos, aun en el medio de otros mensajes MIDI (Fig. 2-28).



**Fig. 2-28.** Los mensajes en tiempo real de sistema pueden insertarse dentro del flujo de bytes de otros mensajes MIDI.

### *Reloj de tiempo*

El mensaje de reloj de tiempo MIDI se transmite dentro del flujo de datos a razón de 24 veces por cuarto de nota (24 ppqn). Este mensaje se usa para sincronizar los relojes de tiempo internos de cada dispositivo MIDI dentro de un sistema y se transmite en los modos de empezar (start) y detener (stop) a la razón de tiempo actualmente definida.

### *Start*

Al recibir un mensaje del reloj de tiempo, el comando MIDI *start* (empezar) instruye a todos los aparatos MIDI conectados a que empiecen a tocar desde el comienzo de su secuencia interna. Si el programa está en medio de la secuencia, el comando *start* repondrá la secuencia desde su comienzo, desde donde empezará a tocar.

### *Stop*

Al recibir un comando *stop* todos los aparatos dentro



del sistema..se detendrán en su actual punto de posición.

### ***Continúe***

Luego de recibir un mensaje MIDI *stop*, un mensaje MIDI *Continúe* instruirá a todos los dispositivos conectados a reanudar la ejecución de sus secuencias internas desde el punto preciso en el cual fueron detenidas.

### ***Sensado activo***

Cuando está en el modo *stop*, puede transmitirse un *mensaje de sensado activo* opcional a través del flujo de datos cada 300 milisegundos. Esto instruye a los dispositivos capaces de reconocer este mensaje que el dispositivo está activamente conectado al sistema MIDI.

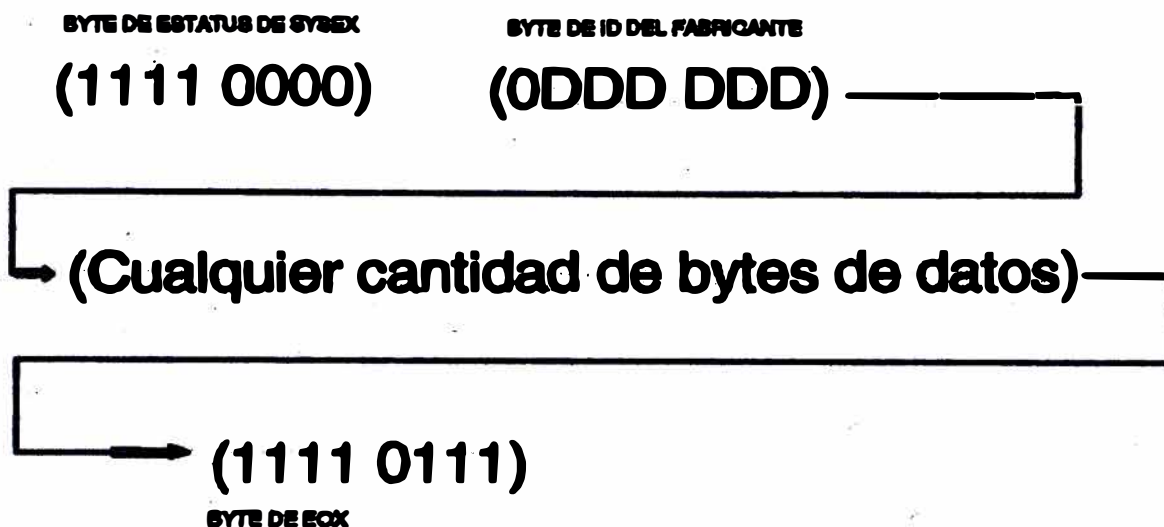
### ***Reinicialización del sistema***

Este mensaje se transmite manualmente en orden a reinicializar un instrumento o dispositivo MIDI a sus posiciones o configuraciones de encendido por defecto (comúnmente Modo 1, control local on, y todas las notas off).

## **MENSAJES EXCLUSIVOS DE SISTEMA**

Los *mensajes exclusivos de sistema* (System-exclusive messages, SysEx) permiten a los fabricantes, programadores y diseñadores MIDI comunicar mensajes MIDI acostumbrados entre dos dispositivos. Es el propósito de estos mensajes dar a los fabricantes, programadores y diseñadores la libertad de comunicar cualesquiera datos de un dispositivo específico de una longitud irrestricta, como sea conveniente. Comúnmente, los datos SysEx se usan para la transmisión y recepción en volumen de datos de

programa, datos simples y control en tiempo real sobre los parámetros de un dispositivo (así como los programas de editor/librería).



**Fig. 2-29.** Datos de Sistema-exclusivo (formato de un byte de ID).

El formato de transmisión para mensajes SysEx (Fig. 2-29) como se define en el estándar MIDI incluye un encabezado de status de SysEx, número ID del fabricante, cualquier número de bytes de datos SysEx y un byte EOX. Al recibir un mensaje SysEx, un dispositivo MIDI lee el número de identificación para determinar si los mensajes siguientes son relevantes o no. Esto se logra fácilmente, ya que un único número ID de uno o tres bytes es asignado a cada fabricante MIDI registrado. Si este número no es el mismo que el dispositivo receptor tiene, los posteriores bytes de datos serán ignorados. Una vez que se transmite un flujo de datos SysEx válidos, se envía un mensaje final EOX, después del cual el dispositivo responderá nuevamente a los mensajes de performance MIDI que ingresen.

### ***Sistema exclusivo universal en tiempo no real***

Los datos SysEx universales en tiempo no real son un protocolo que se usa para comunicar datos de control y de performance en tiempo no real. Se usa actualmente para comunicar en forma inteligente un *protocolo de datos de salud* (informar a un dispositivo sobre un estado o evento específico, como también pedir datos específicos). También se usa para la transmisión y recepción de datos universales de muestreo, y finalmente para transmitir mensajes de apuntes de código de tiempo MIDI. Un mensaje SysEx universal en tiempo no real consta de cuatro o cinco bytes que incluyen dos bytes de datos sub-ID que identifican a cuál parámetro de tiempo no real se debe direccionar. Luego sigue un flujo de datos SysEx pertinentes.

### ***Sistema exclusivo universal en tiempo real***

Actualmente, están definidos dos mensajes SysEx universales en tiempo real. Ambos se relacionan con el código de sincronización MTC. Estos incluyen datos completos de mensaje (relativos a una dirección SMPTE) y datos de bit del usuario.

#### **2.2.6 Status corriente**

Dentro de la especificación MIDI 1.0 se ha tomado especiales provisiones para reducir la necesidad de comunicar datos MIDI redundantes. Este modo, conocido como *status corriente* (running status), permite que una serie de mensajes MIDI consecutivos que tienen el mismo tipo de byte de status se comuniquen sin repetir bytes de

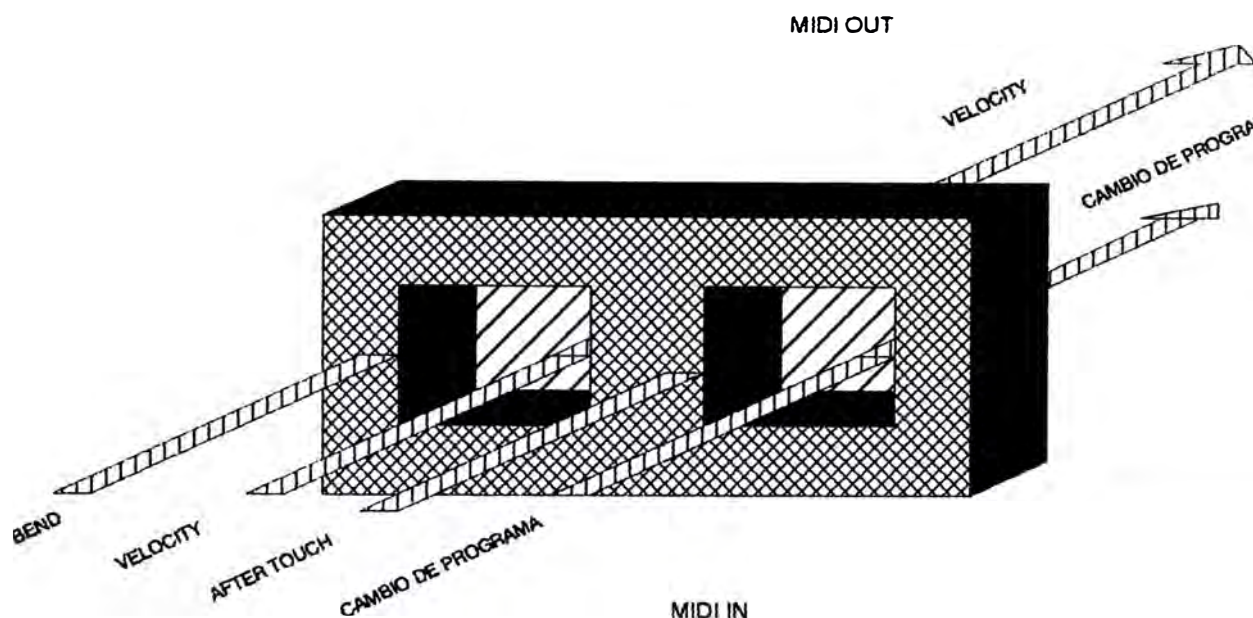
status redundantes cada vez que se envía un mensaje MIDI. Por ejemplo, conocemos que un mensaje MIDI estándar está hecho de un byte de status y uno o más bytes de datos. Cuando se usa el status corriente, sin embargo, una serie de mensajes de curva de pitch (pitch-bend) que han sido generados por un controlador transmitirían un mensaje de byte de status y datos inicial, seguido solo por una serie de bytes datos relacionados (niveles de curva de pitch), sin la necesidad de incluir bytes de status redundantes. Se puede decir lo mismo para note-on, note-off, o cualquier otro tipo de mensaje de status. Al recibir un mensaje que incluye un nuevo valor de byte de status, el dispositivo responderá a él de similar modo, hasta que se encuentre la siguiente serie de bytes de status similares.

Aunque la transmisión de mensajes de status corriente es opcional, todos los dispositivos MIDI deben ser capaces de identificar y responder a este modo de transmisión de datos.

### **2.3 Filtrado MIDI**

Un *filtro MIDI* es un dispositivo digital, procesador o programa de computadora que permite que algunos mensajes o rangos de mensajes dentro de un flujo de datos, sean reconocidos o ignorados. Un filtro de datos MIDI puede pensarse como un switch digital de paso/no-paso que puede programarse para bloquear la transmisión de mensajes específicos MIDI (Fig. 2-30), tales como on/off de velocidad, on/off de cambio de programa, on/off de

modulación, on/off de SysEx, etc.



**Fig. 2-30.** Un filtro MIDI se usa para bloquear la transmisión de mensajes MIDI específicos.

Un instrumento o dispositivo MIDI puede ser capaz de filtrar datos MIDI que ingresan o filtrar datos en sus puertos *out* o *thru*. En este último caso, el dispositivo mismo no será afectado. Sin embargo, todos los dispositivos que siguen en la cadena pueden contener datos filtrados selectivamente. Adicionalmente, es posible filtrar mensajes que sean transmitidos sobre canales MIDI seleccionados por el usuario (afectando así sólo a un patch de instrumento o dispositivo específico dentro del sistema).

#### 2.4 Mapeo MIDI

Un *Dispositivo de mapeo MIDI* es un aparato digital dedicado, procesador o programa de computadora que puede

usarse para reasignar los valores de los bytes de datos o rango de bytes de datos a otros valores de bytes de datos o rango de bytes de datos definidos por el usuario. El mapeo puede aplicarse dentro de la cadena MIDI para reasignar números de canal, números de programa, números de nota (para transportar notas o crear acordes), números y valores de controlador, etc. Al igual que el filtro MIDI, también es posible mapear específicos bytes de mensaje que son transmitidos sobre canales MIDI seleccionados por el usuario.

## **2.5 Resumen**

De la información de este capítulo hemos visto que la actual especificación MIDI 1.0 es un medio digital poderoso para comunicar información de performance y control a través de un sistema musical electrónico.

## **CAPITULO III SISTEMAS DE HARDWARE DENTRO DE LA PRODUCCIÓN MIDI**

Adicionalmente al amplio rango de instrumentos electrónicos MIDI en el mercado, existe también un gran número de sistemas de hardware MIDI que han sido diseñados para ayudar en la distribución y procesamiento MIDI. Tales dispositivos de conexión y hardware incluyen sistemas que proporcionan manejo de datos, procesamiento, distribución e interfase. Ya que el MIDI es un lenguaje estandarizado, sus datos pueden comunicarse a todos los dispositivos MIDI que estén conectados dentro de la cadena de señal para integrarlos en un único y eficiente ambiente de producción.

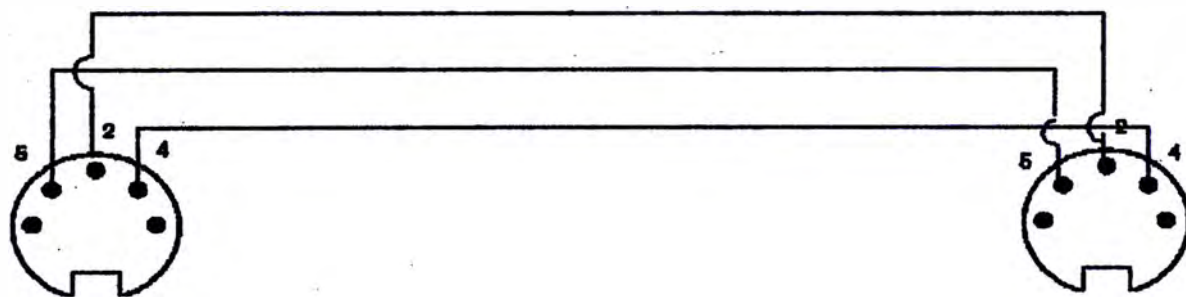
### **3.1 Interconexión del sistema**

Como medio de transmisión de datos, el MIDI es relativamente único dentro del mundo de la producción sonora, permitiendo que 16 canales de datos de performance, controladores y tiempo se transmitan en una dirección sobre un simple cable MIDI. Usando este medio, es posible que los mensajes MIDI se comuniquen a un número de dispositivos dentro de una red desde un controlador maestro (un teclado o secuenciador MIDI) sobre una cadena de datos MIDI.

### **3.2 El cable MIDI**

Un Cable MIDI (Fig. 3-1) consiste en un par de

conductores entramados y blindados con un conector DIN macho de 5 pines en cada uno de sus extremos. La especificación MIDI 1.0 usa sólo 3 de los 5 pines, los pines 4 y 5 se usan para conducir datos y el pin 2 se usa para conectar el blindaje del cable a la tierra del equipo. Los pines 1 y 3 no se usan, pero están reservados para posibles cambios en futuras aplicaciones MIDI. El cable entramado y la conexión a tierra se usan para reducir la interferencia exterior, como radiofrecuencias o interferencias electrostáticas, que pueden distorsionar o corromper la transmisión del mensaje MIDI.



**Fig. 3-1. Diagrama del cable MIDI.**

Los cables MIDI prefabricados, en longitudes de 1, 2, 3, 6 y 15 mts, pueden obtenerse en tiendas de música que vendan equipo MIDI. Sin embargo, 50 pies (15.24 mts) es la máxima longitud especificada por la especificación MIDI para reducir los efectos de degradación de la señal e interferencia externa, que suele ocurrir en recorridos largos de cable.



### 3.2.1 Puertos MIDI

Dentro de la distribución de datos MIDI, existen tres tipos de puertos MIDI que hacen uso de jacks DIN de 5 pines para lograr interconexión entre dispositivos MIDI dentro de una red: MIDI IN, MIDI OUT y MIDI thru (Fig. 3-2). El diseño de hardware para estos puertos (como está estrictamente definido por la especificación MIDI 1.0) incluye el uso de optoaisladores, que sirven para eliminar posibles problemas de lazo de tierra dentro de las líneas de datos conectados MIDI.

#### 3.2.1.1 MIDI IN

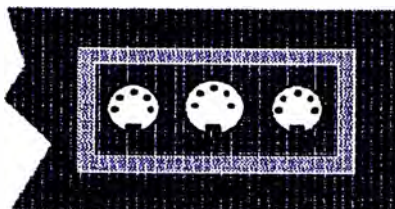
El puerto MIDI IN recibe mensajes MIDI desde una fuente externa y comunica estos datos de performance, control y tiempo al microprocesador interno del dispositivo.

Puede diseñarse más de un puerto MIDI IN en un sistema para proporcionar funciones de  *fusión*  (merge), o para algunos dispositivos MIDI que soportan más de 16 canales. Otros dispositivos (como los controladores MIDI) pueden no requerir el uso de un puerto MIDI IN.

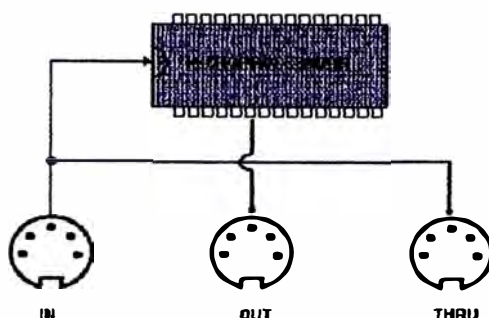
#### 3.2.1.2 MIDI OUT

El puerto MIDI OUT se usa para transmitir mensajes MIDI desde una fuente única al microprocesador de otro instrumento o dispositivo MIDI.

Se puede diseñar más de un puerto MIDI OUT en un sistema para el simple propósito de proporcionar múltiples salidas MIDI (proveyendo distribución del mismo flujo de datos a una cantidad de instrumentos).



(A) Puertos MIDI detrás de un dispositivo MIDI



(B) Camino de la señal de los puertos MIDI in, out y thru

### **Fig. 3-2. Puertos MIDI in, out y thru.**

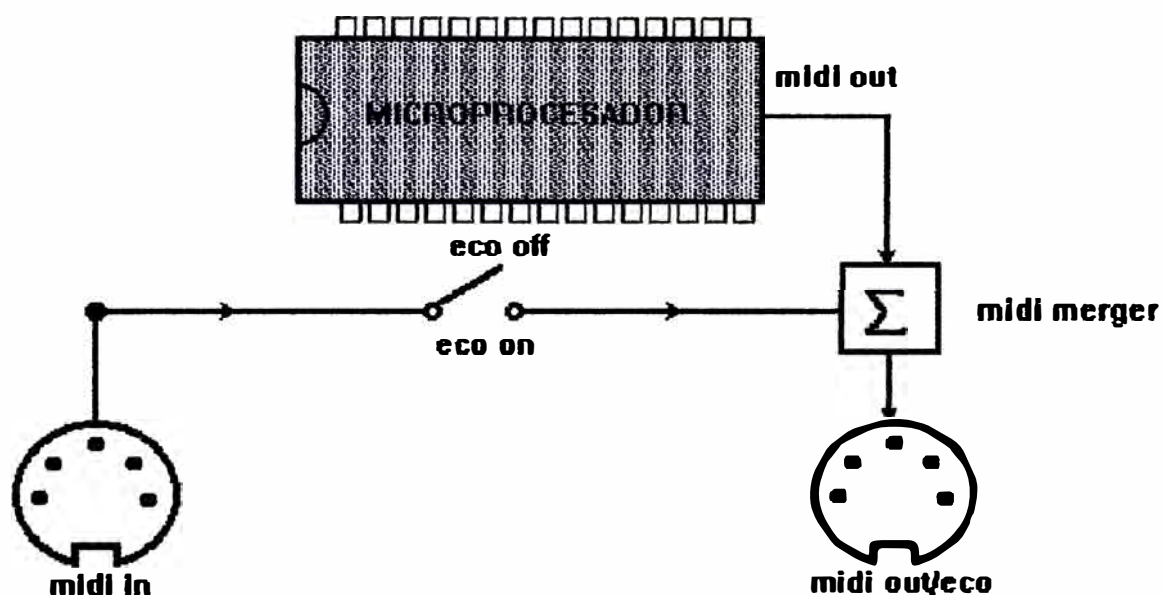
Alternativamente, los dispositivos que pueden soportar más de 16 canales frecuentemente tienen la habilidad de rutear información de canal de datos MIDI a más de un puerto aislado MIDI. Esto tiene la ventaja de proporcionar un sistema con mayores capacidades por canal (proveyendo aislamiento de datos entre puertos MIDI y reduciendo posibles obstaculizaciones de datos), y permitir que el usuario filtre datos MIDI sobre otro puerto, mientras no esté restringiendo selectivamente el flujo de datos dentro de otro puerto.

#### **3.2.1.3 MIDI THRU**

El puerto MIDI thru proporciona una copia exacta de los datos de entrada en el puerto MIDI IN. Se le usa para transmitir estos datos a otro instrumento o dispositivo

MIDI que esté a continuación dentro de nuestra cadena de datos MIDI. Este puerto se usa para entregar una copia exacta del flujo de datos del MIDI IN en el MIDI thru, sin fusionarlo con los datos que se transmiten en el puerto MIDI OUT.

### 3.2.2 Eco MIDI



**Fig. 3-3.** Configuración de eco MIDI.

Ciertos dispositivos no incluyen un puerto MIDI thru. Estos dispositivos, sin embargo, pueden ofrecer una función de transmisión basada en un software, para seleccionar entre un puerto MIDI OUT y un puerto *MIDI echo* (Fig. 3-3). Tal como con el puerto MIDI thru, la función MIDI echo seleccionable se usa para proporcionar una copia exacta de cualquier información que sea recibida en el puerto MIDI IN y rutear estos datos al puerto MIDI OUT/echo. A diferencia de un puerto MIDI OUT

dedicado, la función MIDI echo puede frecuentemente seleccionarse para fusionar datos de entrada con datos de performance que son generados por el mismo dispositivo. De esta manera, más de un controlador puede simultáneamente ser colocado dentro de un sistema MIDI. Es importante notar que aunque los datos de tiempo y performance son comunicados por todos los dispositivos que hacen uso del MIDI echo, no todos los dispositivos pueden hacer eco con datos SysEx.

### 3.2.3 Configuraciones típicas:

Aunque existe una gran variedad de tipos de dispositivos y diseños de sistema de una configuración MIDI a otra, hay ciertas convenciones que permiten a los aparatos MIDI conectarse fácilmente dentro de un sistema MIDI. Estas configuraciones comunes permiten que los datos MIDI se distribuyan de la manera más eficiente posible.

Como regla principal, hay sólo dos métodos válidos para conectar un dispositivo MIDI a otro (Fig 3-4):



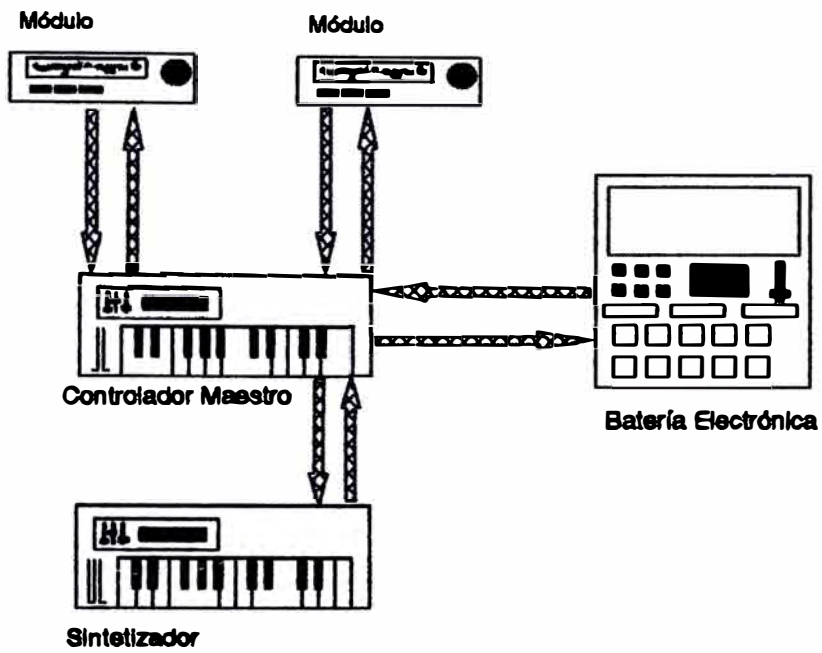
**Fig. 3-4.** Las dos maneras válidas de conectar un dispositivo MIDI a otro.

puede entonces continuar a lo largo de un sistema básico MIDI hasta que se alcanza el aparato final.

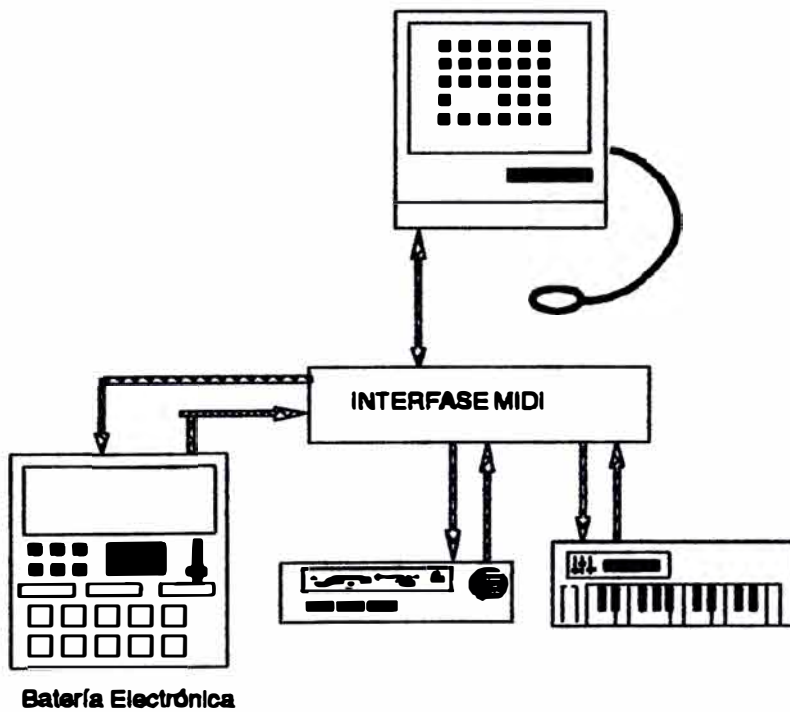
Puesto que el puerto MIDI thru sólo pasará datos que son copia exacta de los datos en su puerto MIDI IN, la señal puede ser rastreada a través de cada dispositivo hasta un controlador maestro único (tal como un teclado). En la mayoría de los casos, esto es aceptable porque el controlador se usa para transmitir datos sobre uno o más canales MIDI que, a su turno, le serán individualmente respondidos por dispositivos que han sido asignados a estos canales. Cualquier dispositivo dentro de una cadena puede usarse como una fuente de control si simplemente se conecta su puerto MIDI OUT en el puerto MIDI IN de cualquier dispositivo que venga a continuación dentro de la cadena MIDI.

### **3.2.3.2 La Red Estrella**

Otro método para conectar múltiples aparatos MIDI dentro de un sistema es a través del uso de la *Red Estrella* (Fig 3-6). Este tipo de estrategia de interconexión permite que un controlador maestro se comunique con un determinado número de instrumentos MIDI (o cadenas de instrumentos) sobre puertos MIDI individualmente direccionables. El ruteo de datos con tal red permite que cada juego de puertos MIDI IN y OUT acomode flujos de datos MIDI aislados independientemente. Esto nos da la mayor flexibilidad en las interconexiones del sistema.



(A) Interconexiones básicas del sistema



(B) Uso común con la nueva generación de Interfaces MIDI

**Fig. 3-6. La red Estrella**

Dentro de sistemas MIDI más grandes y complejos, una Red Estrella ofrece ciertas ventajas sobre la Cadena Margarita. Permite que un controlador, capaz de direccionar cada rama de una red, comunique datos (vía los puertos MIDI IN y OUT) que son relevantes a cada instrumento o cadena de instrumentos. Como los 16 canales de datos MIDI pueden ser transmitidos y recibidos sobre cada rama, este tipo de red es también ideal para usarse con dispositivos MIDI que puedan manejar más de 16 canales (P. ej., un programa de secuencia de 32 canales).

Los paneles de conexiones MIDI (*MIDI Patchbay*), que son aparatos capaces de enrutar selectivamente los datos MIDI, además de generaciones nuevas de interfases de computadora, están frecuentemente equipados con múltiples puertos MIDI IN y OUT, para facilitar la creación de la Red Estrella.

### **3.2.3.3 Transmisión inalámbrica MIDI**

Están saliendo al mercado algunos sistemas MIDI que permiten la transmisión de mensajes MIDI desde una posición remota a un receptor estacionario que está conectado al sistema MIDI, con las ventajas consecuentes.

El *TransMidi* de la firma *MidIMAN* (Fig. 3-7) es un dispositivo que permite que la mayoría de sistemas inalámbricos puedan convertirse en sistemas inalámbricos MIDI. Consiste en dos módulos: un módulo receptor y un transmisor. El transmisor codifica datos MIDI estándar en una señal de audio que puede transmitirse usando un transmisor inalámbrico estándar. Esta señal de audio

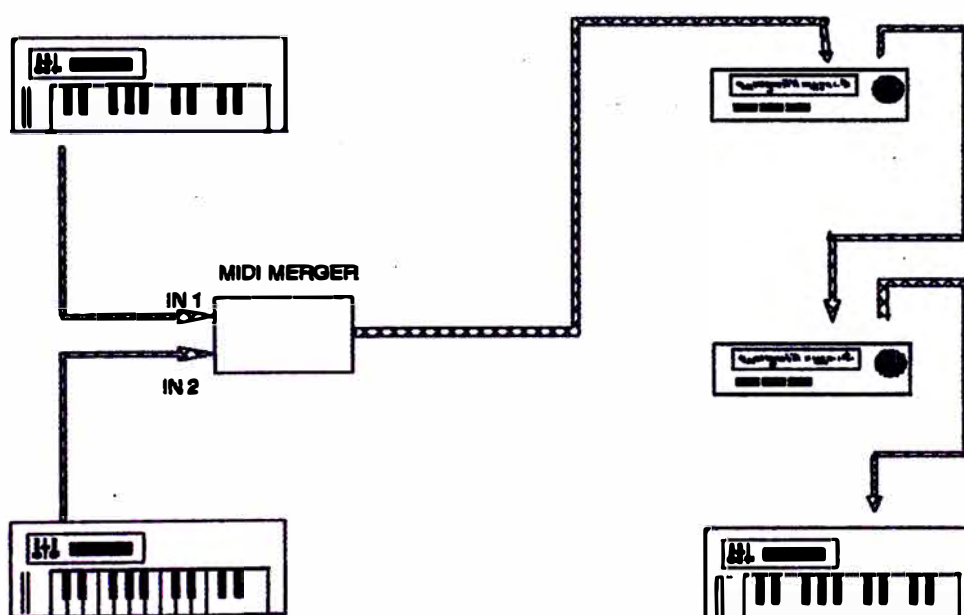


codificada se recibe luego usando un receptor inalámbrico convencional y es reconvertida usando el receptor *TransMidi* en la información MIDI original.

### 3.3 Hardware de manejo de datos

Existe en el mercado un amplio rango de dispositivos de Hardware MIDI que están diseñados para el propósito de distribuir y manejar datos MIDI dentro de un sistema. Ejemplos de estos dispositivos son: el MIDI merger, MIDI thru box, MIDI switcher, MIDI patchbay y herramientas de diagnóstico y procesadores de datos MIDI.

#### 3.3.1 MIDI Merger



**Fig. 3-8.** Ejemplo de un MIDI Merger dentro de un sistema.

Cuando se combinan los datos de dos o más líneas MIDI en un solo flujo de datos MIDI, debe usarse un *MIDI merger* (Fig. 3.8). Esto debe hacerse porque no es posible

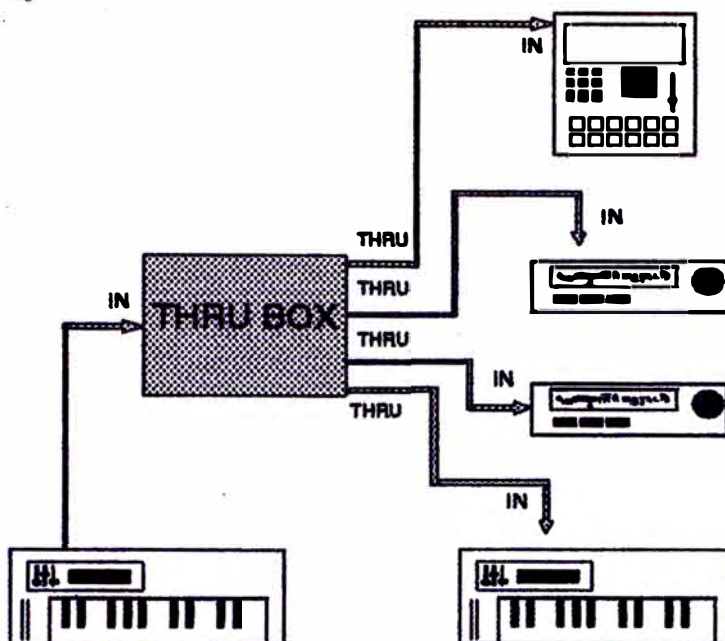


juntar simplemente dos líneas de datos MIDI en una entrada, ya que la colisión de datos resultante sería inevitablemente irreconocible como datos MIDI válidos. En cambio, el trabajo de un MIDI Merger es intercalar las líneas de datos ingresantes en un flujo MIDI continuo que contenga mensajes MIDI válidos de cada fuente MIDI.

Una función del MIDI merger puede ser hacer que dos dispositivos MIDI sirvan de controladores maestros dentro de un sistema MIDI. Por ejemplo, en la Figura 3-8, los datos MIDI OUT de ambos teclados o de un controlador de viento MIDI son fusionados (*merge* en inglés) en una sola línea para que así cualquiera pueda ser usado para controlar un sistema o dispositivo externo.

### **3.3.2 MIDI Thru Box**

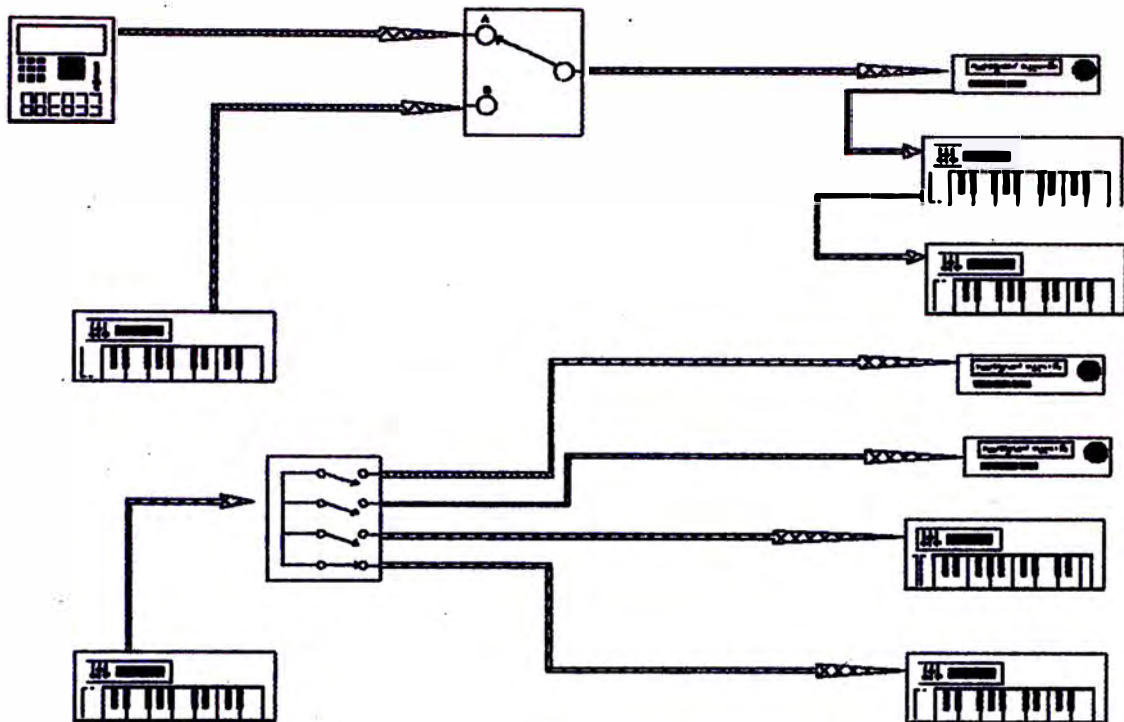
Una *MIDI Thru Box* se usa para distribuir una fuente de datos MIDI a través de un sistema, ya que provee una copia exacta de los datos de entrada a una cantidad de puertos MIDI THRU (Fig 3-9). Tal caja (box) puede usarse para reemplazar las conexiones MIDI THRU dentro de una Cadena Margarita, ya que efectivamente conecta cada dispositivo a una copia exacta de la fuente de datos. Esto se hace usualmente para evitar retrasos en la velocidad de transmisión que ocurren dentro de cadenas de datos que consisten en cuatro o más dispositivos MIDI.



**Fig. 3-9.** Ejemplo de una MIDI Thru Box dentro de un sistema.

### 3.3.3 MIDI Switcher

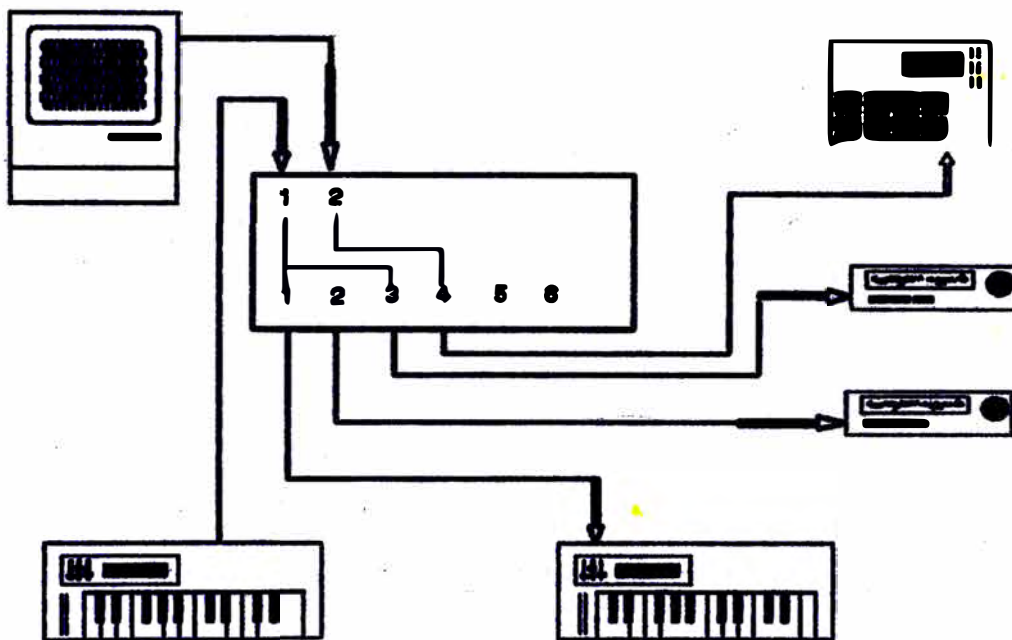
Un switcher MIDI permite al usuario seleccionar entre dos o más fuentes controladores MIDI sin la necesidad de reconectar manualmente en el patch los cables MIDI. La figura 3-10 nos muestra cómo un switcher MIDI puede usarse para seleccionar entre un teclado y una batería electrónica dentro de un sistema.



**Fig. 3-10.** Ejemplos de un Switcher MIDI dentro de un sistema.

### 3.3.4 MIDI Patchbay

Siempre que una cantidad de dispositivos esclavos MIDI están bajo el control de uno o más controladores MIDI, es frecuentemente útil emplear un *MIDI Patchbay* (Panel de conexiones MIDI). La función del MIDI patchbay es encaminar selectivamente los senderos de señal de datos MIDI dentro de un sistema de producción. Tal dispositivo puede usarse para enrutar una fuente controladora MIDI a uno o más dispositivos MIDI dentro de un sistema conectado (Fig. 3-11). Alternativamente, múltiples controladores pueden ser seleccionados para que controlen simultáneamente dispositivos MIDI separados dentro de un sistema.



**Fig. 3-11.** Diversos ejemplos de un Patchbay MIDI dentro de un sistema.

Los sistemas de panel de conexiones MIDI están disponibles con un amplio rango de configuraciones de puertos y opciones. Ciertos sistemas requieren que el usuario conmute manualmente la configuración del sistema, mientras otros permiten al usuario llamar configuraciones que están preseteadas.

Un ejemplo de un sistema básico es el patchbay MIDI MX-28S (2 in y 8 out) de la Digital Music Corp. (Fig. 3-12). Este dispositivo permite al usuario seleccionar el flujo de datos MIDI desde dos dispositivos de control independientes a cualquiera de ocho dispositivos esclavos MIDI. Esto se logra incorporando un interruptor de tres posiciones para cada puerto de salida usado para seleccionar cualquiera de las posibles entradas como fuente de control para cada puerto MIDI de salida. Cuando el interruptor está en la posición central off, no se distribuirán datos al puerto de salida seleccionado.

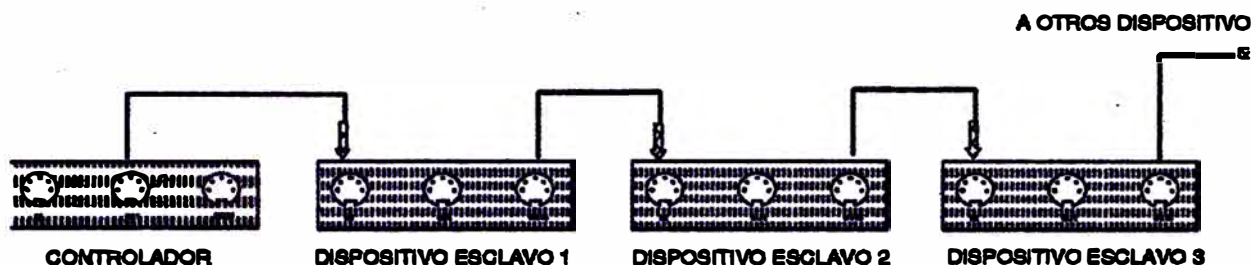
### **3.3.5 Conversor MIDI / Trigger**

Otro dispositivo interesante es el Conversor MIDI/trigger. Trigger es un término electrónico en inglés que significa pulso de disparo. Algunos instrumentos de percusión electrónica, como las baterías Simmons, Roland DDR30 o Boss, tienen salidas Trigger (es decir que generan un pulso de disparo que se necesita para activar el aparato). El conversor tiene varias entradas para trigger programables para dirigirse desde varios aparatos. Se le asigna a cada trigger un número de nota MIDI y un canal MIDI. Así, cuando el conversor recibe un

1.- Conectar el puerto MIDI OUT (o puerto MIDI OUT/echo) de un dispositivo al puerto MIDI IN de otro dispositivo.

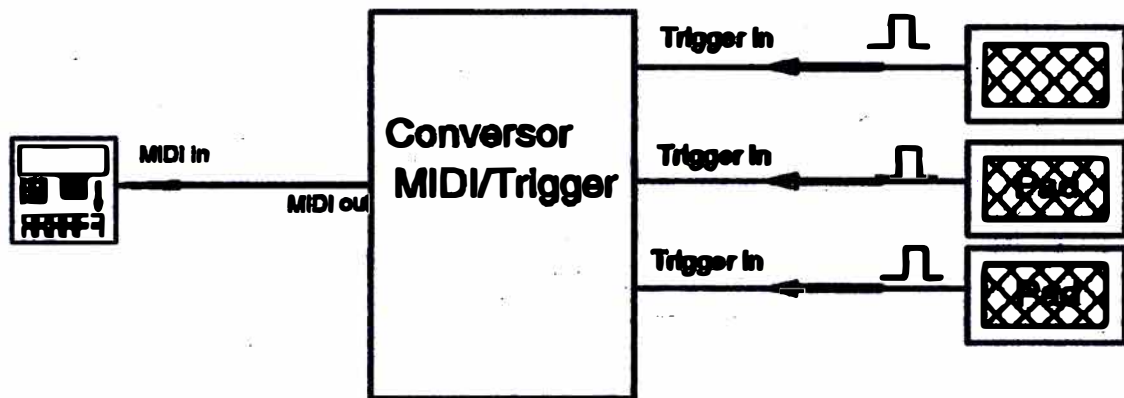
2.- Conectar el puerto MIDI thru de un aparato al puerto MIDI IN de otro aparato.

### 3.2.3.1 La Cadena Margarita (Daisy Chain)



**Fig. 3-5.** Ejemplo de un sistema conectado MIDI usando la cadena margarita.

Uno de los métodos más simples y más comúnmente usados para distribuir datos dentro de un sistema es la *MIDI daisy chain*. Este método se usa para distribuir una línea de datos MIDI simple a todos los aparatos dentro de un sistema al transmitir datos al primer dispositivo y subsecuentemente pasar una copia exacta de estos datos a través de cada dispositivo dentro de la cadena (Fig. 3-5). Esto se hace al enviar los datos MIDI OUT del aparato fuente (controlador, secuenciador, etc.) al puerto MIDI IN del segundo dispositivo. Al conectar el puerto MIDI thru del segundo dispositivo al puerto MIDI IN del tercer aparato, este último recibirá una copia exacta de los datos de fuente originales en esta entrada. Este proceso



**Fig. 3-13.** Conversor de disparos (trigger) a datos MIDI.

pulso, está ordenando al conversor enviar una nota, canal e información dinámica MIDI correspondiente a la asignación de entrada MIDI correspondiente. Esto es útil, por ejemplo, cuando un baterista quiere tener acceso a sonidos sintetizados de batería usando tambores de trigger (Fig. 3-13). Este modo de operación se denomina *Conversión Trigger/MIDI*. El proceso inverso también es posible, aunque actualmente es menos frecuente porque ya casi no se usan los sonidos de las baterías electrónicas antiguas.

### **3.3.6 Procesadores de datos MIDI**

Así como los procesadores se pueden usar en una cadena de audio para crear efectos al cambiar o enriquecer un sonido existente, un procesador MIDI puede colocarse dentro de una cadena de datos MIDI para encaminar o alterar los mensajes MIDI dentro del sistema.

Un procesador MIDI es frecuentemente capaz de desarrollar un amplio rango de funciones algorítmicas sobre una o más señales MIDI, tales como:

- \* Reasignación de canal MIDI y controlador continuo.

- \* Delay programable MIDI.

- \* Filtrado de mensaje MIDI

- \* Limitación y escala de velocidad



\* Transposición cromática.

\* Inversiones de nota, velocidad y rueda de afinación.

Un ejemplo de un sistema de procesamiento MIDI multifuncional es el procesador *MIDIbuddy multi-MIDI* de la ACME Digital Inc. Este dispositivo presenta 10 puertos MIDI IN independientes y 10 puertos MIDI OUT (proveyendo panel de conexiones y control de ruta sobre un total de 160 canales), procesamiento MIDI (incluyendo MIDI merge, delay, filtrado y mapeo), registro de datos SysEx (para almacenar patch universal y otros datos específicos del dispositivo), y capacidades de Grabación/Reproducción directamente de una secuencia de diskette.

El MIDIbuddy permite al usuario automatizar el patch y las funciones de configuración del sistema (por medio de mensajes de control o disparados en tiempo real desde un controlador MIDI remoto). En adición a un amplio rango de capacidades de procesamiento, este dispositivo es capaz de almacenar secuencias MIDI y datos SysEx en uno o dos de los diskettes de 3.5" del aparato o a un disco duro externo o computadora por medio de un puerto SCSI opcional.

### **3.3.7 Herramientas de diagnóstico MIDI**

Aunque existe una gran número de herramientas basadas en software para analizar datos MIDI a través del uso de la computadora, hay también disponibles herramientas de hardware para diagnosticar datos MIDI dentro de un

estudio de producción MIDI o una puesta en escena en vivo. Tales herramientas se usan frecuentemente para detectar la presencia de datos MIDI o para revertir algún mensaje de error específico.

Un ejemplo de una herramienta simple de diagnóstico es el *MIDI Beacon* de Musonix (Fig. 3-14). Este dispositivo de bolsillo se usa para indicar la presencia de datos MIDI dentro de un sistema. Esto se logra por medio de un *Led* que brilla cuando detecta presencia de datos MIDI. Este simple artefacto puede usarse como luz indicadora de datos para reordenar datos en una cadena interconectada o para chequear cables MIDI.

El analizador *MA36 MIDI* de Studiomaster Inc. (Fig. 3-15) es una herramienta de mano que indica el canal MIDI usado dentro de una red y muestra el tipo de mensaje MIDI que está siendo transmitido o recibido. Las conexiones MIDI IN y THRU permiten que este analizador sea colocado dentro de la cadena de datos.

### **3.3.8 La red de área local (LAN)**

Cuando grandes volúmenes de canales, notas, SysEx, y otros datos se embuten a través de una sola línea de datos seriales a 31.25 kilobaudios dentro de un sistema de producción MIDI a gran escala, nos estamos limitando a correr en un tráfico que frecuentemente derivará en retrasos MIDI. Gran parte de estos problemas de configuración de sistema y distribución están actualmente siendo direccionados por nuevas generaciones de interfases MIDI, patches y sistemas de procesamiento. Sin

embargo, uno de estos sistemas de administración MIDI está ganado reconocimiento al combatir los problemas de configuración de sistema y distribución MIDI a través del uso de una *red de área local* (local area network, LAN) (Fig 3-16), una tecnología ampliamente usada dentro del mundo de los negocios. Una de las mayores ventajas de una LAN es que permite que una multitud de computadoras separadas (o en este caso, de sistemas MIDI) se conecten directamente, permitiéndoles intercambiar datos en tiempo real a una razón de transferencia mucho mayor de lo que sería vía MIDI.

El primero de estos sistemas LAN para uso dentro de la producción MIDI es el MidiTap de Lone Wolf. El MidiTap es un dispositivo que ocupa un espacio de Rack que tiene un panel frontal que incluye cuatro botones de control de software, un pequeño display de cristal líquido y un panel de parámetros. Su panel trasero presenta cuatro puertos MIDI IN y OUT, un puerto serial RS232/422 para conexiones directas a cualquier computadora o modem y un juego de puertos de fibra óptica. Son estos puertos de fibra óptica los que hacen a este aparato tan especial, porque hacen uso del protocolo MediaLink, propio de Lone Wolf.

El MediaLink es un protocolo bidireccional de alta velocidad que facilita la eficiente distribución de gran cantidad de información digital (tal como MIDI, audio digital, SMPTE, etc.) en tiempo real y no real, entre MidiTaps conectados o dispositivos compatibles.

A diferencia del MIDI, que está limitado a una velocidad de 31.25 k-baudios, el MediaLink es capaz de transmitir datos digitales en ambas direcciones a velocidades de 1, 2, 4, 10 y 100 megabits por segundo. Es esta alta velocidad de transferencia de datos la que permite que múltiples MidiTaps se ligen en una red LAN virtual, permitiendo que cualquier dispositivo MIDI conectado a cualquier MidiTap sea direccionado por cualquier otro en la red en una ejecución en tiempo real.

### **3.4 La computadora personal en la producción MIDI**

La computadora personal es frecuentemente un componente central dentro de la mayoría de sistemas MI. A través del uso de programas y periféricos, es usual procesar datos relativos a configuración y performance desde una posición de control integrada.

Hay básicamente tres tipos de computadora personal que son las más comúnmente usadas dentro de la producción MIDI: *Macintosh*, *IBM* (e *IBM* compatibles) y la familia *Atari*. Cada una de ellas ofrece un amplio rango de sistemas de cómputo, sus propias formas y funciones, y sus propias y distintas ventajas y desventajas para el cómputo personal.

La mayor ventaja de la computadora personal es que es una máquina de procesamiento digital de alta velocidad que permite que el usuario/artista adicione las opciones de periféricos y hardware que mejor se adecuan a sus necesidades de producción. Están disponibles en gran variedad de opciones de memoria RAM (por medio de la

expansión de chips de memoria y discos flexibles o duros, o discos ópticos). Una de las mejores ventajas es la variedad de software disponible para cada sistema de cómputo. Es este factor el que convierte a la PC en un camaleón digital (esto es, capaz de cambiar de función en ordena a adecuarse a las tareas necesarias en esa momento). Adicionalmente, la computadora permite que el usuario escoja los sistemas de software que mejor se acomodan a sus necesidades de producción y a su estilo personal de producción.

En este trabajo sólo trataremos el caso de la PC IBM y compatibles.

#### **3.4.1 IBM/compatible**

Debido a su economía, aceptación y uso extendido, se acepta generalmente que hay más computadoras personales IBM/compatibles (Fig. 3-17) en el mercado (musical o de cualquier otra clase), que de cualquier otro tipo. Existe también una gran variedad de computadoras IBM/compatibles y un vasto mercado de hardware y periféricos junto con software para uso en todo tipo de aplicaciones personales y de negocios.

Aunque no podemos mencionar fabricantes y opciones específicas por ser muy variadas, existen cuatro categorías básicas de sistemas IBM/compatibles, que usan los procesadores 80286, 80386, 80486 y Pentium. Cada sistema hace uso de varias versiones del Sistema Operativo de Disco (Disk operating system, DOS, de la Microsoft) como un sistema operativo estandarizado para

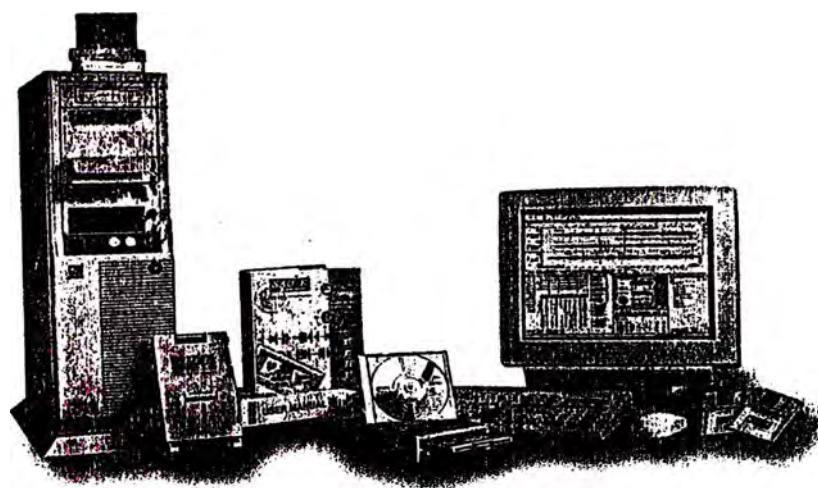


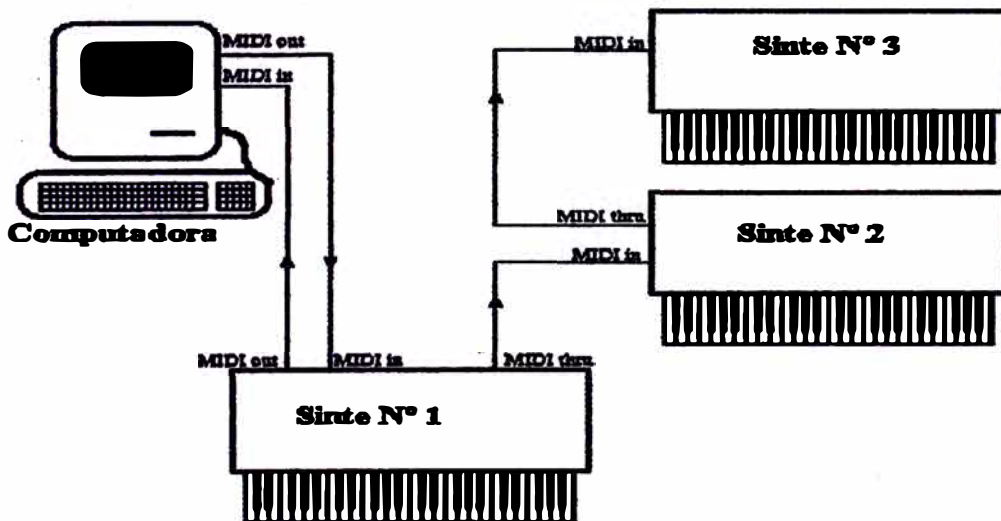
Fig. 3-17. Ejemplo de computadora IBM Compatible

realizar tareas básicas de procesamiento y administración.

Agregando una tarjeta de sonido adecuada, cualquier computadora IBM/compatible básica puede convertirse en un estudio de grabaciones digital con una amplia variedad de opciones, canales de grabación e inclusive con sonidos propios para uso con los secuenciadores.

## CAPITULO IV SECUENCIACION

Las computadoras se usan para secuenciar. El término *Secuenciador* describe un dispositivo usado como grabadora digital. Una computadora sirve como grabadora digital porque puede almacenar mensajes binarios MIDI y luego reproducirlos en el orden o *secuencia* en que fueron recibidos. La Fig. 4-1. muestra una configuración típica en la que una computadora se convierte en un secuenciador.



**Fig. 4-1.** Computadora usada como secuenciador

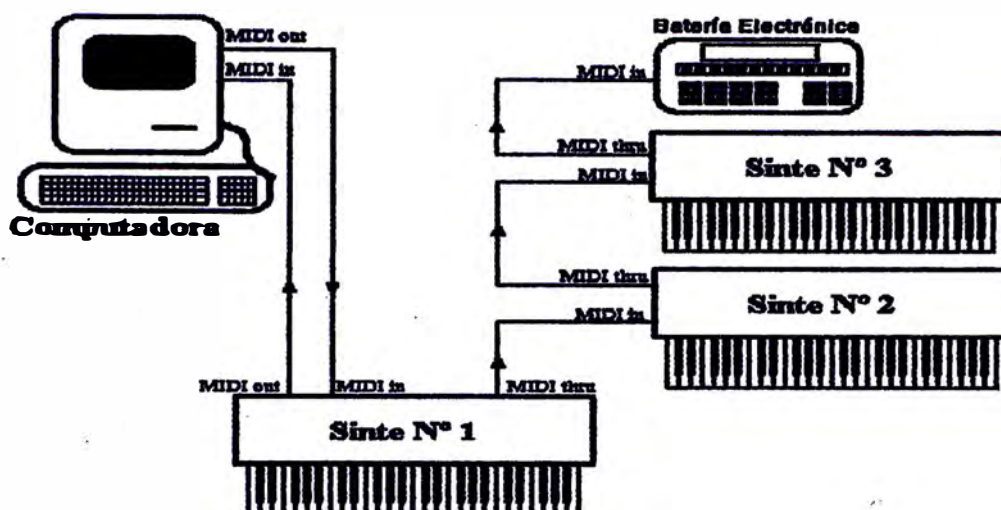
Por ejemplo, digamos que un músico quiere tocar el sintetizador #1 como controlador que envía información a la computadora y entonces pone a la computadora a reproducir esta información a través de los tres



sintetizadores. Primero, debe cargarse un programa de secuenciador en la computadora (sea desde un diskette o desde el disco duro. Al poner la computadora en el modo de grabación, se debe oír un pulso (click) de metrónomo. Cuando el músico toca, la computadora recibe y almacena mensajes MIDI que caracterizan la performance de las teclas tocadas. Después que la grabación termina, la computadora va al modo de reproducción. Los mensajes MIDI son reproducidos, recibidos por el sintetizador #1 y luego enviados nuevamente a los sintetizadores #2 y #3. Algunas ventajas interesantes aparecen aquí. Durante la reproducción el músico puede cambiar programas de sonido y explorar posibilidades. Puede cambiarse el *tempo*, para hacer la reproducción más rápida o más lenta. Inclusive puede transponerse la reproducción a una escala diferente.

Se puede expandir la configuración añadiendo un sintetizador de percusión, que también es un secuenciador, como se muestra en la Fig. 4-2. Asumiendo que ya existe un programa de batería en la máquina, se le puede oír en la reproducción. Los secuenciadores usan códigos especiales llamados *mensajes en tiempo real del sistema*. Cuando se inicia la reproducción, la computadora envía un mensaje MIDI de inicio (start) que tiene valor sólo para la batería electrónica. Esta ahora necesita alguna información del tempo, así que la computadora le envía una serie continua de pulsos llamados MIDI clock (reloj), para sincronizar la batería electrónica con la

reproducción de la computadora. Un comando de parada (stop) termina la reproducción y un comando *continúe*, reinicia la reproducción.



**Fig. 4-2.** Batería electrónica añadida a la cadena MIDI.

Un sintetizador de percusión (batería electrónica), puede tocarse a través de un teclado, puesto que puede responder a ciertas notas o teclas, dependiendo del fabricante. Usualmente hay una provisión en la máquina de ritmos para alterar las asignaciones de teclas para el bombo, tarola, hihats, etc. De hecho, es útil saber el estándar industrial de los números de teclas MIDI. Este estándar ahora se ha generalizado a muchos campos de las producciones MIDI y se le conoce como *General MIDI*. Las notas empiezan desde Do como número 1, luego, el Do central vendría a ser el número 60. Generalmente los teclados comerciales tienen sólo 5 octavas, empezando aproximadamente desde el número 36 (Do). Al ser el número de tecla un número de 7 bits, se puede tener teóricamente un teclado de 128 teclas, es decir, de más de 10 octavas.

## APPENDICE A

# **T H E M I D I 1 . 0 S P E C I F I C A T I O N**

### **INTRODUCTION**

MIDI is the acronym for Musical Instrument Digital Interface.

MIDI enables synthesizers, sequencers, home computers, rhythm machines, etc. to be interconnected through a standard interface.

Each MIDI-equipped instrument usually contains a receiver and a transmitter. The receiver receives messages in the MIDI format and executes MIDI commands. It consists of an opto-isolator, Universal Asynchronous Receiver / Transmitter (UART), and other hardware needed to perform the intended functions. The transmitter originates messages in MIDI format, and transmits them by way of a UART and line driver.

The MIDI standard hardware and data format are defined in this specification.

### **CONVENTIONS**

Status and Data bytes given in Tables I through VI are given in binary.

Numbers followed by an "H" are in hexadecimal.

All other numbers are in decimal.

**HARDWARE-**

The interface operates at 31.25 ( $\pm 1\%$ ) Kbaud, asynchronous, with a start bit, 8 data bits (D0 to D7), and a stop bit. This makes a total of 10 bits for a period of 320 microseconds per serial byte.

Circuit: 5mA current loop type. Logical 0 is current ON. One output shall drive one and only one input. The receiver shall be opto-isolated and requires less than 5 mA to turn on. Sharp PC-900 and HP 6N138 opto-isolators have been found acceptable. Other high speed opto-isolators may be satisfactory. Rise and fall times should be less than 2 microseconds.

Connectors: DIN 5 pin (180 degree) female panel mount receptable. An example is the SWITCHCRAFT 57 GB5F. The connectors shall be labelled "MIDI IN" and "MIDI OUT". NOTE that pins 1 and 3 are not used and should be left unconnected in the receiver and transmitter.

Cables shall have a maximum length of fifty feet (15 metres), and shall be terminated on each end by a corresponding 5-pin DIN male plug, such as the SWITCHCRAFT 05GM5M. The cable shall be shielded twisted pair, with the shield connected to pin 2 at both ends.

A "MIDI THRU" output may be provided if needed, which provides a direct copy of data coming in MIDI IN. For very long chain lengths (more than three instruments), higher speed opto-isolators must be used to avoid additive rise/fall time errors which affect pulse width duty cycle.

## **DATA FORMAT**

All MIDI communication is achieved through multi-byte "messages" consisting of one Status byte followed by one or two Data bytes, except Real-Time and Exclusive messages (see below).

## **MESSAGE TYPES**

Messages are divided into two main categories: Channel and System.

### **CHANNEL**

Channel messages contain a four-bit number in the Status byte which address the messages specifically to one of sixteen channels. These messages are thereby intended for any units in a system whose channel number matches the channel number encoded into the Status byte.

There are two types of Channel messages: Voice and Mode.

Voice- To control the instrument's voices, Voice messages are sent over the Voice Channels.

Mode- To define the instruments response to Voice messages, Mode messages are sent over the instrument's Basic Channel.

### **SYSTEM**

System messages are not encoded with channel numbers

There are three types of System Messages: Common, Real-Time, and Exclusive.

Common-Common messages are intended for all units in a system.

Real-Time-Real-Time messages are intended for all units in a system.

They contain Status bytes only-no Data bytes. Real-Time messages may be sent at any time-even between bytes of a message which has a different status. In such cases the Real-Time message is either ignored or acted upon, after which the receiving process resumes under the previous status.

Exclusive-Exclusive messages can contain any number of Data bytes, and are terminated by an End of Exclusive (EOX) or any other Status byte. These messages include a Manufacturer's Identification (ID) code. If the ignore the ensuing data.

So that other users can fully access Midi instruments, manufacturers should publish the format of data following their ID code. Only the manufacturer can update the format following their ID.

### ***DATA TYPES***

#### **STATUS BYTE**

Status byte are eight-bit binary numbers in which the Most Significant Bit (MSB) is set (binary 1). Status bytes serve to identify the message type; that is, the purpose of the Data bytes which follow the status byte. Except for Real-Time messages, new Status bytes will always command the receiver to adopt their status, even if the new Status is received before the last message was completed.

Running Status-For Voice and Mode messages only, when a Status byte is received and processed, the receiver will remain in the status until a different Status byte is

received... Therefore, if the same Status byte would be repeated. It may (optionally) be omitted so that the only correct number of data bytes need be sent. Under Running Status, then, a complete message need only consist of Specified Data bytes sent in the specified order.

The Running Status feature is specially useful for communicating long strings of Note On/Off messages, where "Note On with Velocity of 0" is used for Note Off. (A separate Note Off Status byte is also available).

Running Status will be stopped when any other Status byte intervenes, except the Real-Time messages will only interrupt the Running Status temporarily.

Unimplemented status-Any Status byte received for functions which the receiver has not implemented should be ignored, and subsequent data bytes ignored.

Undefined Status-Undefined Status bytes must not be used. care should be taken to prevent illegal messages from being sent during power-up or power-down. If undefined Status bytes are received, they should be ignored, as should subsequent Data bytes.

#### **DATA BYTES**

Following the Status byte, there are (except for Real-Time messages) one or two Data bytes which carry the content of the message. Data bytes are eight-bit binary numbers in which the MSB is reset (binary 0). The number and range of Data bytes which must follow each Status byte are specified in the tables which follow. For each Status byte the correct number of Data bytes must always

be sent. Inside the receiver, action in the message should ignore Data Bytes which have not been properly preceded by a valid Status byte (with the exception of "Running Status", above).

### ***CHANNEL MODES***

Synthesizers contain sound generation elements called voices. Voice assignment is the algorithmic process of routing Note On/Off data from the keyboard to the voices so that the musical notes are correctly played with accurate timing.

When MIDI is implemented, the relationship between the sixteen available MIDI channels and the synthesizer's voice assignment must be defined. Several Mode messages are available for this purpose (see Table III). They are Omni (On/Off), Poly, and Mono. Poly and Mono are mutually exclusive, i.e., Poly Select disables messages in Voice Channels without discrimination. When Omni is off, the receiver will accept Voice messages from only the selected Voice to just one voice per voice Channel (Monophonic). When Mono is off (=Poly On), any number of voices may be allocated by the Receiver's normal voice assignment algorithm (polyphonic).

For a receiver assigned to Basic Channel "N", the four possible modes arising from the two Mode messages are:



MODE	CHNI		
1	On	Poly	Voice messages are received from all Voice Channels and assigned to voices polyphonically.
2	On	Mono	Voice messages are received Channels, and control only one voice, monophonically.
3	Off	Poly	Voice messages are received in Voice Channel N only, and are assigned to voices polyphonically.
4	Off	Mono	Voice messages are received in Voice Channels N thru N+M-1, and assigned monophonically to voices 1 thru M, respectively. The number of voices M is specified by the third byte of the Mono Mode Message.

Four modes are applied to transmitters (also assigned to Basic Channel N). Transmitters with no channels selection capability will normally transmit on Basic Channel 1 (N=0).

MODE	OMNI		
1	On	Poly	All voice messages are transmitted in Channel N.
2	On	Mono	Voice messages for one voice are sent in Channel N.
3	Off	Poly	Voice messages for all voices are sent in Channel N.
4	Off	Mono	Voice messages for voice 1 thru M are transmitted in Voice Channels N thru N+m-1, respectively. (Single voice per channel.)

A MIDI receiver or transmitter can operate under one and only one mode at a time. Usually the transmitter will be in the same mode. If mode cannot be honoured by the receiver, it may ignore the message (and any subsequent data bytes), or it may switch to an alternate mode (usually Mode 1, Omni On/Poly).

Mode messages will be recognized by a receiver only when sent in the Basic Channel to which the receiver has been assigned, regardless of the current mode. Voice messages may be received in the Basic Channel and in other channels (which are all called Voice Channels), which are related specifically to the Basic Channel by the rules above, depending on which mode has been selected.

A MIDI receiver may be assigned to one or more Basic Channels by default or by user control. For example, an eight-voice synthesizer might be assigned to Basic Channel 1 on power-up. The user could then switch the instrument to be configured as two four-voice

synthesizers, each to its own Basic Channel. Separate Mode messages would then be sent to each four-voice synthesizer, just as if they were physically separate instruments.

***POWER-UP DEFAULT CONDITIONS***

On power-up all instruments should default to Mode # 1. Except for Note On/Off Status, all Voice messages should be disabled. Spurious or undefined transmissions must be suppressed.

TABLE I. SUMMARY OF STATUS BYTES

STATUS	# OF DATA	DESCRIPTION
D7-D0	BYTES	
		Channel Voice Messages
1000nnnn	2	Note Off event.
1001nnnn	2	Note On event (velocity=0:Note Off).
1010nnnn	2	Polyphonic key pressure/after touch.
1011nnnn	2	Control change.
1100nnnn	1	Program change.
1101nnnn	1	Channel pressure/after touch.
1110nnnn	2	Pitch bend change.
		Channel Mode Messages
1011nnnn	2	Selects Channel Mode.
		System Messages
11110000	1	System Exclusive.
11110sss	0 to 2	System Common.
11111ttt	0	System Real Time.

## Notes:

nnnn: N-1, where N=Channel #,  
 i.e., 0000 is Channel 1.  
 0001 is Channel 2.  
 .  
 .  
 .  
 1111 is Channel 16.

01111111, data, . . . , BOX.  
 11111111: Identification.

sss: 1 to 7.  
 ttt: 0 to 7.

TABLE II. CHANNEL VOICE MESSAGES

STATUS	DATA BYTES	DESCRIPTION
1000mmmm	0kkkkkkk 0vvvvvvv	Note Off (see notes 1-4) vvvvvvv note off velocity
1001mmmm	0kkkkkkk 0vvvvvvv	Note On (see notes 1-4) vvvvvvv=0: velocity vvvvvvv=0: note off
1010mmmm	0kkkkkkk 0vvvvvvv	Polyphonic Key Pressure (After-Touch) vvvvvvv: pressure value
1011mmmm	0ccccccc 0vvvvvvv	Control Change ccccccc: control # (0- 121)(see 5-8) vvvvvvv: control value ccccccc=122 thru 127: Reserved.(See Table III)
1100mmmm	0FFFFFFF	Program Change FFFFFFF: program number (0- 127)
1101mmmm	0vvvvvvv	Channel Pressure (After- Touch) vvvvvvv: PRESSURE VALUE
1110mmmm	0vvvvvvv 0vvvvvvv	Pitch Bend Change LSB (see note 10) Pitch Bend Change MSB

## Notes:

1. mmmm: Voice Channel # (1-16, coded as defined in  
Table I notes)

2. kkkkkk: note # (0-127)

kkkkkk=60: Middle C of keyboard

0 12 24 36 48 60 72 84 96 108 120 127

ac c c c c c c c

|-----piano range-----|

3. **vvvvvvv: key velocity**

A logarithmic scale would be advisable.

0      1                                  64    127

---

off    ppp    pp    p    mp    mf    f    fff    fff

vvvvvvv=64: in case of no velocity sensors

vvvvvvv=0: Note Off, with velocity = 64

4. Any Note On message sent should be balanced by sending a Note Off message for that note in that channel at some later time.

5. **ooooooo: control number**

<b>OOOOOOO</b>	<b>DESCRIPTION</b>
0	Continuous Controller 0 MSB
1	Continuous Controller 1 MSB (MODULATION BENDER)
2	Continuous Controller 2 MSB
3	Continuous Controller 3 MSB
4-31	Continuous Controller 4-31 MSB
32	Continuous Controller 0 LSB
33	Continuous Controller 1 LSB (MODULATION BENDER)
34	Continuous Controller 2 LSB
35	Continuous Controller 3 LSB
36-63	Continuous Controller 4-31 LSB
64-95	Switches (On/Off)
96-121	Undefined
122-127	Reserved for Channel Mode messages (see Table III).

6. All controllers are specifically defined by agreement of the MIDI Manufacturers Association (MMA) and the Japan MIDI Standards Committee (JMSC). Manufacturers can request through the MMA or JMSC that logical controllers be assigned to physical ones as necessary. The controller allocation table must be provided in the user's operation manual.

7. Continuous controllers are divided into Most Significant and Least Significant Bytes. If only seven of resolution are needed

for any particular controllers, only the MSB is sent. It is not necessary to send the LSB. If more resolution is needed, then both are sent, first the MSB, then the LSB. If only the LSB has changed in value, the LSB may be sent without re-sending the MSB.

8. vvvvvv: control value (MSB)

(for controllers)

0-----127

min

max

(for switches)

0- - - - - 127

off

on

Numbers 1 through 126, inclusive, are ignored.

9. Any messages (e.g., Note On), which are sent successively under the same status, can be sent without a Status byte until a different Status byte is needed.
- 10 Sensitivity of the pitch bender is selected in the receiver. Center position value (no pith change) is 2000H, wich would be transmitted EnH-00-40H.

TABLE III. CHANNEL MODE MESSAGES

STATUS	DATA BYTES	DESCRIPTION
10110	ccccccc 0vvvvvvv	Mode Messages  ccccccc=122: Local Control vvvvvvv=0, Local Control Off vvvvvvv=127, Local Control On ccccccc=123: All Notes Off vvvvvvv=0 ccccccc=124: Omni Mode Off (All Notes Off) vvvvvvv=0 ccccccc=125: Omni Mode On (All Notes Off) vvvvvvv=0 ccccccc=126: Mono Mode On (Poly Mode Off) (All Notes Off) vvvvvvv=M, where M is the number of channels. vvvvvvv=0, the number of channels equals the number of voices in the receiver. ccccccc=127: Poly Mode On (Mono Mode Off) vvvvvvv=0 (All Notes Off)

## Notes:

1.nnnn: Basic Channel # (1-16, coded as defined in Table I)

2.Messages 123 thru 127 functions All Notes Off messages. They will turn off all voices controlled by the assigned Basic Channel. Except for message 123, All Notes Off, they should not be sent periodically, but only for a specific purpose. In no case should they be used in lieu of Note Off commands to turn off notes



which have been previously turned on. Therefore any All Notes Off command (123-127) may be ignored by receiver with no possibility of notes staying on, since any Notes staying on, since any Note On command must a corresponding specific Note Off command.

3. Control Change # 122, Local Control, is optionally used to interrupt the internal control path between the keyboard, for example, and the sound generating circuitry. If 0 (Local Off message) is received, the path is disconnected: the keyboard data goes only to MIDI and the sound generating circuitry is controlled only by incoming MIDI data. If a 7FH (Local On message) is received, normal operation is restored.

4. The third byte of "Mono" specifies the number of channels in which Monophonic Voice messages are to be sent. This number of channels in which Monophonic Voice messages are to be sent. This number, "M", is a number between 1 and 16. The channel(s) being used, then, will be the current Basic Channel (=N) thru N+M-1 up to a maximum of 16. If M=0, this is a special case directing the receiver to assign all its voices, one channel, from the Basic Channel N through 16.

TABLE IV. SYSTEM CONTROL MESSAGES

STATUS	DATA BYTES	DESCRIPTION
11110001		Undefined
11110010		Song Position Pointer
	01111111	11111111:(Least significant)
	0hhhhhhh	hhhhhhh: (Most significant)
11110011	0sssssss	Song Select
		sssssss: Song #
11110100		Undefined
11110101		Undefined
11110110	none	Tune Request
11110111	none	EOX: "End of System Exclusive" flag.

1. Song Position Pointer: Is internal register which holds the number of MIDI beats (1 beat=6 MIDI clocks) since the start of the song. Normally it is set to 0 when the START switch is pressed, which starts sequence playback. It then increments with every sixth MIDI clock receipt, until STOP is pressed. If CONTINUE is pressed, it continues to increment. It can be arbitrarily preset (to a resolution of 1 beat) by the SONG POSITION POINTER message.
2. Song Select: Specifies which song or sequence is to be played upon receipt of a Start (Real-Time) message.
3. Tune Request: Used with analog synthesizers to request them to tune their oscillators.
4. EOX: Used as a flag to indicate the end of a System Exclusive transmission (see Table VI).

**TABLE V. SYSTEM REAL TIME MESSAGES**

<b>STATUS</b>	<b>DATA BYTES</b>	<b>DESCRIPTION</b>
11111000		Timing Clock
11111001		Undefined
11111010		Start
11111011		Continue
11111100		Stop
11111101		Undefined
11111110		Active Sensing
11111111		System Reset

**Notes:**

1. The System Real Time messages are for synchronizing all of the system in real time.
2. The System Real Time messages can be sent at any time. Any messages which consist of two or more bytes may be split to insert Real Time messages.
3. Timing clock (FSH)  
The system is synchronized with this clock, which is sent at a rate of 24 clocks/quarter note.
4. Start (from the beginning of song) (FAH)  
This byte is immediately sent when the PLAY switch on the master (e.g., sequencer or rhythm unit) is pressed.
5. Continue (FBH)  
This is sent when the CONTINUE switch is hit. A sequence will continue at the time of the next clock.
6. Stop (FOH)  
This is immediately sent when the STOP switch is hit. It will stop the sequence.
7. Active Sensing (FEH)  
Use of this message is optional, for either receivers or transmitters. This is a "dummy" Status byte that is sent every 300 ms (max), whenever there is no other activity on MIDI. The receiver will operate normally if it never receives FEH.

Otherwise, if FEH is ever received, the receiver will expect of 300 ms passes with no activity, the receiver will turn off the voices and return to normal operation.

**B. System Reset (FFH)**

This message initializes all of the condition of just having turned on power. The system Reset message should be used sparingly, preferably under manual command only. In particular, it should not be sent automatically on power up.

TABLE VI. SYSTEM EXCLUSIVE MESSAGES

STATUS	DATA BYTES	DESCRIPTION
11110000	01111111 (0-127)	Bulk dump etc. 1111111: identification.
	(0-127)	Any number of bytes may be sent here, for any purpose, as they all have a zero in the most significant bit.
	11110111	EOX: "End of System Exclusive"

## Notes:

1. 1111111: identification ID (0-127)
2. All bytes between the System Exclusive Status byte and EOX or the next Status byte must have zeroes in the MSB.
3. The ID number can be obtained from the MMA or JMBC.
4. In no case should other Status or Data bytes (except Real-Time) be interleaved with System Exclusive, regardless of whether or not the ID code.

## APPENDICE B

# T H E M I D I I M P L E M E N T A T I O N C H A R T

It is always necessary for a MIDI device to receive every type of MIDI message that is defined by the MIDI specification. Certain messages may not to the function of a device. For example, it is only required of a synth module that it respond to note-on/off messages since there is no built-in keyboard controller for transmitting them. Other devices might limit various MIDI messages due to such factors as design limitations or cost-effectiveness. For instance, should a velocity-sensitive keyboard controller be used to control a synthesizer that does not respond to velocity messages, no amount of keyboard banging will result in a change in volume. As a result, the velocity message is simply.

To insure that two or more MIDI devices will be able to communicate MIDI events effectively, the MMA and the JMCS have devised the *MIDI Implementation Chart* (Fig B-1), which relates all of the MIDI capabilities of specific MIDI device to the user at a glance using a standardized printed format.

From the user's standpoint, when considering a new piece of equipment, it is always wise to compare its implementation chart with other devices existing messages and/or add to the capabilities of the current system.

***GUIDELINES FOR USING THE CHART***

The MMA specifies that the MIDI implementation chart be printed the same size using a standardized spreadsheet format consisting of 4 columns by 12 rows. The first column lists the MIDI function in question. The second lists information relating to whether (or how) the device recognizes (receives) this data, and the final column is used for additional remarks by manufacturer.

Function...		Transmitted	Recognized	Remarks
Basic Channel	Default Changed	1 - 16 1 - 16	1 - 16 1 - 16	recognized
Mode	Default Messages Altered	Mode 3,4 OMNI OFF, MONO POLY *****	Mode 3 Y	recognized
Note Number	True Voice	8-127 *****	8 - 127 12 - 100	
Velocity Note ON	Note OFF	⊙ v = 1 - 127 x 9n v = 0	⊙ v = 1 - 127 Y	
After Touch	Key's Ch's	Y Y	Y Y	
Pitch Bender		⊙	⊙ 0 - 24 semitons	
Control Change	1	⊙	⊙	Modulation
	2-6	Y	Y	
	4	ss	ss	Data Entry MSB Values
	7	⊙	⊙	
	8-10	Y	Y	Genl Purpose Ctrl-1
	14	Y	⊙	
	17-37	Y	Y	
	38	ss	Y	Data Entry LSB
	39-63	Y	Y	
	64	Y	⊙	Hold 1
	65-99	Y	Y	
	91	Y	⊙	Genl Purpose Ctrl-1
	92-99	Y	Y	
100-101	ss(0)	ss (0)	RPC LSB, MSB	
102-120	Y	Y		
121	⊙	⊙	Reset All Controllers	
Prog Change	True 0	⊙ 0 - 127 *****	⊙ 0 - 127 0 - 127	
System Exclusive		⊙	⊙	
System Common	Song Pos Song Sel Tune	Y Y Y	Y Y Y	
System	Clock Real Time Commands	Y Y	Y Y	
Aux Message	Local ON/OFF All Notes OFF Active Sense Reset	Y Y ⊙ Y	Y ⊙ ⊙ Y	
Notes:		<p>3 Control Change messages from 8 to 99 which are recognized through Control channel are transmitted through all the channels which are used in branches. However, General Purpose Control - 1 and General Purpose Control - 4 are converted into the same functions as the FC-100 EV-0 assign and the PC-100 Switch assign in the System Setup, and are transmitted.</p> <p>ssRPC = Registered Parameter Control Number RPC 0 0 : Bender Range The value of parameter is to be determined by entering data.</p>		

Mode 1: OMNI ON, POLY    Mode 2: MONI ON MONO  
 Mode 3: MONI OFF, POLY    Mode 4: OMNI OFF, MONO

⊙: Yes  
 Y: No

Fig. B-1 Example of a MIDI Implementation Chart.



Despite efforts at standardization, slight inconsistencies within the chart's specifications allow for variations in the symbols, abbreviations, spelling, etc. that can be used by different manufacturers. The following guidelines a basic understanding of these differences.

- In general, the symbol O is used to indicate that a MIDI function *is* implemented, while an X is used to show that the function is *not* implemented. However, some charts may use an X to equal a *yes* and an O to equal a *no*. This will usually be indicated within a key at the lower right-hand corner of the chart.
- OX or "\*" is used to indicate a selectable function. Further information on the range or type of selectability will be placed within the remark column.
- MIDI modes are listed as follows:
  - Mode 1 (omni on, poly)
  - Mode 2 (omni on, mono)
  - Mode 3 (omni off, poly)
  - Mode 4 (omni off, mono)

These will be listed at the bottom of the chart. Occasionally abbreviations of these (i.e., omni on/off, omni on or poly) may be used by a manufacturer.

## **DETAILED EXPLANATION OF THE CHART**

The following is a detailed explanation of various functions and their related categories that are found within the chart.

### **Header**

The *header* provides the user with the model number, brief description, date, and version number of the device.

### **Basic Channel**

*Basic channel* indicates which MIDI channels are used by the device to transmit and receive data. The subheadings for this function are *default* and *change*.

- *Default:* This indicates which MIDI channel is in use when the device is first turned on.
- *Changed:* This indicates which of the MIDI channels can be addressed after the device is first turned on.

### **Mode**

*Mode* indicates which of the MIDI modes may be used by the device. The subheading for this function are *default*, *messages*, and *altered*.

- *Default:* This indicates which of the MIDI mode is activated when the device is first turned on.
- *Messages:* This describes which of the four MIDI modes can be transmitted or recognized by the device.
- *Altered:* This refers to mode messages which cannot be recognized by the device. It may be followed by a description of the mode that the

device automatically enters into upon receiving a request message for an unavailable mode.

### **Note Number**

The transmitted *note number* indicates the range of MIDI note numbers that are transmitted by a device. The maximum possible range spans 0-127, while 21-108 corresponds to the 88 keys of an extended keyboard controller. Should the note number be greater than the number of keys on keyboard device, a key transposition feature is indicated.

The recognized note number indicates the range of MIDI note numbers that can be recognized by a device. MIDI notes that are out of this range shall be ignored by this device. A second note number range, known as *true voice*, indicates the number of notes the device can actually play. Recognized notes that are out of the actual voice range are transposed up or down in octaves until they fall within this range.

### **Velocity**

This category indicates whether the device is capable of transmitting or receiving attack -and release-velocity messages. The subheadings for this function are *note on* and *note off*.

- *Note on*: This indicates if the device is capable of transmitting and responding to variable-velocity (attack) messages. Not all dynamically controllable devices respond to the full velocity range (1-127). Some devices, such

as drum machines, respond to a finite number of velocity steps.

- *Note off*: Indicates whether the device is capable of transmitting and responding to variable release velocity messages. Many devices use a message (note-on velocity=0) to indicate a note-off condition. This is often indicated in the chart by *9NH v=0* or *\$9n 00*, which is the hexadecimal equivalent for this message.

### **After Touch**

*After touch* indicates how pressure data is transmitted or received. The subheadings for this function are *key's* and *ch's*.

- *Key's*: This indicates if the device will transmit or receive independent polyphonic-pressure messages for each key.
- *Ch's*: Indicates whether the device is capable of transmitting or receiving channel-pressure changes (a common after touch mode, proving one pressure value for an entire MIDI channel)

### **Pitch Bender**

*Pitch bender* indicates if the device is capable of transmitting or receiving pitchbend information. If so, the remarks column will often give information as the

pitch bend range and resolution.

### **Control Change**

*Control Change* indicates whether the device is capable of transmitting or receiving continuous-controller messages. The chart will often lists which of these messages are supported in addition to providing a detailed breakdown of their parameters within the remarks column.

### **Program Change**

This category indicates if the device is capable of transmitting or receiving *program-change messages*. *True #* indicates the message numbers that are actually supported by the device's program-change buttons.

### **System Exclusive**

This indicates if the device is capable of transmitting and receiving *system exclusive data*. The remark column will often give general information as to which type of SysEx data is supported. However, more detailed data will generally be provided within the device's manual.

### **System Common**

This indicates whether the device is capable of transmitting or receiving the different types of *system common messages*, such as SPP, MIDI time code, song select, and tune-request messages.

### **System Real Time**

This category indicates whether the device can transmit or receive *system real-time messages*. The subheadings for this function are *clock* and *commands*.

- *Clock*: This refers to the device's ability to receive or transmit MIDI clock messages. A device that can transmit MIDI clock may be used to provide master timing information within a MIDI system, while a device capable of receiving clock data may only be slaved to other MIDI devices.
- *Commands*: This indicates whether the device is capable of transmitting or responding to start, stop, and continue messages.

### **Aux Messages**

This indicates if a device is capable of transmitting or receiving local control-on/off, all notes-off, active-sensing, and system-reset messages.

### **Notes**

This area is used by the manufacturer to comment on any function or implementation particular to the specific MIDI device.

## CONCLUSIONES

Hemos intentado presentar esta tecnología novedosa a un nivel diferente al que se ha tratado en los últimos años. Su uso extendido en nuestro medio hace que sea necesario investigar a fondo el contenido del lenguaje MIDI para poder manejar los dispositivos a ese nivel con absoluta independencia tecnológica. Existe ya una industria musical MIDI bastante considerable en nuestro país, esperamos que la lectura de este trabajo lleve a técnicos e ingenieros a investigar y profundizar en el manejo de la interfase MIDI y así la producción musical en nuestro país tendrá el apoyo tecnológico que tienen países avanzados y que los ha llevada a una excelente calidad de producción. Como vimos en el capítulo 1, son múltiples los campos en los que puede aplicarse esta tecnología, esperamos que el conocimiento del hardware y software MIDI lleve a nuestros ingenieros a cierta especialización en este campo que aún no ha sido explotado comercialmente a nivel de ingeniería. La implementación de un circuito de experimentación MIDI viene a ser de lo más elemental. Una simple PC y un teclado económico con puertos MIDI puede servir para empezar a experimentar y mostrar en vivo la tecnología a los alumnos de la UNI. La implementación de un pequeño laboratorio MIDI podría ser

un paso adicional para llevar a nuestro centro de estudios a abrir su campo de acción a la aplicación de la electrónica a un área no explotada hasta el momento. Así como existe la electrónica médica, la electrónica industrial, y otras especializaciones en las que la electrónica se mezcla con otras áreas del conocimiento, a través de este trabajo queremos presentar la especialización de ELECTRÓNICA MUSICAL, carrera que en otros países toma diferentes nombres, como Ingeniería Musical, o Music Engineering Technology, etc. Espero que este trabajo sea de vuestro interés.



## BIBLIOGRAFIA

### Capítulo I:

- 1.- Craig Anderton. *MIDI for Musicians*, New York, NY, ASmsco Publications, 1986.
- 2.- Helen y Frederick David Casabona. *Using MIDI*. Cupertino, CA, GPI Publications, 1987.

### Capítulo II:

- 3.- Steve De Furia y Joe Scacciaferro, *MIDI Programmer's Handbook*, Redwood City, CA, M&T Publishing, Inc., 1989.
- 4.- Michael Boom, *Music Through MIDI*, Redmond, WA, Microsoft Press, 1987.
- 5.- Helen y Frederick David Casabona. *Using MIDI*. Cupertino, CA, GPI Publications, 1987.

### Capítulo III:

- 6.- Steve De Furia y Joe Scacciaferro, *MIDI Programmer's Handbook*, Redwood City, CA, M&T Publishing, Inc., 1989.
- 7.- Dennis N. Nardantonio. *Sound Studio Production Techniques*, Blue Ridge Summit, PA, TAB Books, 1990.
- 8.- Michael Boom, *Music Through MIDI*, Redmond, WA, Microsoft Press, 1987.

### Capítulo IV:

- 9.- Dennis N. Nardantonio. *Sound Studio Production Techniques*, Blue Ridge Summit, PA, TAB Books, 1990.

**Lectura adicional recomendada:****Revistas:**

- 1.- Electronic Musician  
P.O. Box 41094  
Nashville, TN 37204  
USA
- 2.- Keyboard  
P.O. Box 58528  
Boulder, CO 80322  
USA
- 3.- International Musician & Recording World  
Stonehart Subscription Services  
Hainault Road, Little Heath  
Romford, Essex, England RM6 5NP
- 4.- International MIDI Association Bulletin  
5316 West 57th Street  
Los Angeles, CA 90056  
USA
- 5.- Home & Studio Recording (US)  
Music Maker Publications Inc.  
22024 Lassen Street, Suite 118  
Chatsworth, CA 91311  
USA
- 6.- Modern Drummer  
P.O. Box 480  
Mount Morris. IL 61054  
USA
- 7.- Computer Music Journal  
Journals Department  
The Massachusetts Institute of Technology Press  
55 Hayward Street  
Cambridge, MA 02142  
USA