

**UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA EN COMPUTACION**



**DISEÑO DE UNA APLICACIÓN DE CAPTURA Y REPRODUCCION DE NOTAS
MUSICALES EN FORMATO MIDI.**

**TRABAJO DE GRADUACION PARA OPTAR AL GRADO DE INGENIERO EN
CIENCIAS DE LA COMPUTACION.**

PRESENTADO POR:

Daniel Amilcar Romagoza Turcios RT960176

Director de Escuela:

Lic. Mauricio Coto.

Tutor:

Ing. Jaime Anaya.

Asesor:

Ing. Carlos Puente.

por un número de bytes de datos más arriba que el número normal se asoció con ese byte de estado. Los bytes de datos adicionales se asumen para ser asociados al byte de estado pasado recibido —y ellos se procesan apenas como si el octeto de estado había sido repetido.

Aquí está un ejemplo de cómo los mensajes de canal de presión descritos arriba Pueden ser enviados usando Runing Status. (Los mensajes están en el canal 1, y todo los números se demuestran en hexadecimal)

D0 01 02 03 04... 7D 7E 7F 7E 7D... 03 02 01 00

Si no para el Runing Status, el transmisor tendría que enviar esta versión más larga del mismo mensaje.

D0 01 D0 02 D0 03 D0 04... D0 7D D0 7E D0 7F D0 7E D0 7D. .
D0 03 D0 02 D0 01 D0 00

Aquí está otro ejemplo. Bajo Runing Status, enviando los valores 40H, 32H, 26H, 14H, y 00 al regulador 1 en el canal 3 podían ser logrados por la secuencia de datos siguiente.

B2 01 40 01 32 01 26 01 14 01 00

Esto será interpretado como si fuera enviada como lo siguiente.

B2 01 40 B2 01 32 B2 01 26 B2 01 14 B2 01 00

AGREDECIMIENTOS

- A mi Dios todopoderoso que me dio la fuerza, inteligencia y sabiduría que necesitaba, por guiarme en todo el proceso de trabajo de graduación permitiéndome finalizar con éxito el presente proyecto.

- A Sofía, mi querida esposa; por haber sido mi apoyo incondicional, mi respaldo en todo momento, por haberme tenido la paciencia suficiente cuando no tuve el tiempo para estar juntos, por brindarme el amor y la fuerza para seguir adelante cuando ya no podía; pero sobre todo por haberme ayudado tanto y ser como la compañera de tesis que siempre quise tener.

- A toda mi familia por estar conmigo siempre, dándome la fuerza y confianza en todo momento.

- Y al final dedico este triunfo a mis padres, por inculcarme los valores morales que necesitaba para llegar a ser un hombre de bien.

Este triunfo alcanzado es fruto de mi gran dedicación y esfuerzo como persona.

Gracias Señor Jesús por el éxito tan anhelado que he obtenido!!

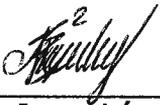
DANIEL AMILCAR ROMAGOZA TURCIOS

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA EN COMPUTACION

JURADO EVALUADOR DEL PROYECTO DE GRADUACION

**DISEÑO DE UNA APLICACIÓN DE CAPTURA Y REPRODUCCION DE NOTAS
MUSICALES EN FORMATO MIDI.**

Firmas:



Ing. Jorge López

Jurado



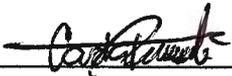
Lic. Oscar Méndez

Jurado



Ing. Arnoldo Rivas

Jurado



Ing. Carlos Puente

Asesor



Ing. Jaime Anaya

Tutor



Daniel Romagoza

Estudiante

ÍNDICE

INTRODUCCIÓN.....	1
--------------------------	----------

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1. OBJETIVOS	2
1.1. OBJETIVO GENERAL.....	2
1.2.OBJETIVOS ESPECIFICOS.....	2
2. ANTECEDENTES DEL TEMA O PROBLEMA	3
3. IMPORTANCIA Y JUSTIFICACIÓN DEL TEMA.....	5
4. PROYECCIÓN SOCIAL.....	6
5. DEFINICION DEL TEMA	7
6. ENFOQUE CON QUE SE ABORDARA EL TEMA	8
7. ALCANCES.....	9
8. LIMITACIONES	11

CAPITULO II

ESTUDIO DE FACTIBILIDAD

1. FACTIBILIDAD TÉCNICA	12
2. FACTIBILIDAD ECONÓMICA	26
3. FACTIBILIDAD OPERATIVA.....	28

**CAPITULO III
METODOLOGÍA DE LA INVESTIGACION**

1. METODOLOGIA DE LA INVESTIGACIÓN.....	30
2. TECNICAS A UTILIZAR EN LA INVESTIGACIÓN.....	32
3. INVESTIGACION PRELIMINAR Y RECOLECCION DE INFORMACIÓN ...	33

**CAPITULO IV
DESCRIPCIÓN DEL PROYECTO**

1. SITUACION ACTUAL	34
2. RESULTADOS ESPERADOS	34
3. DIFICULTADES TECNICAS ENCONTRADAS	34

**CAPITULO V
MARCO TEORICO**

1. MARCO HISTORICO.....	36
2. MARCO CONCEPTUAL.....	41

**CAPITULO VI
DESARROLLO DEL DISEÑO**

1. DISEÑO DEL SISTEMA.....	67
1.1.DISEÑO DE MENUS	67
1.2.DISEÑO DE PANTALLAS	72
1.3.ESTRUCTURA DE ARCHIVOS.....	75

CONCLUSIONES	75
BIBLIOGRAFIA.....	77
GLOSARIO DE TERMINOS.....	78

ANEXOS

<u>1. MANUAL DE USUARIO</u>	87
<u>2. MANUAL DEL PROGRAMADOR.</u>	120

INTRODUCCIÓN

En menos de 15 años, MIDI (Interfaz Digital de Instrumento Musical) ha tomado el mundo de la música como una tormenta. Las interfaces MIDI (el Hardware que hace a MIDI disponible para un instrumento o computadora) aparecen sobre los Teclados musicales, tarjetas de sonido, tarjetas madres de la computadora, guitarras, grabadores de tape, violines y todo lo que se refiera a música digital.

El primer nombre de MIDI es música y la música es su primer enfoque, usando un programa secuenciador, podemos grabar, editar y guardar música en un archivo de disco. Mientras que cada secuenciador generalmente usa su propio formato de archivo propietario, mas secuenciadores tambien soportan un formato común: el formato SMF (formato de archivo Midi estandar). Este archivo de formato estándar permite ser compartidas las composiciones en varios secuenciadores. Tal archivo de formato estandar es comúnmente disponible en el internet a traves de las páginas web y sitios ftp. Grabar y compartir composiciones MIDI ha llegado ser un popular hobby para los musicos. Usando un Web-Browser plug-in hace la tarea de tocar canciones midi en tiempo real.

Los programas MIDI pueden tener un gran uso en instituciones musicales, usando apropiadamente el entrenamiento musical a traves de la computadora, haría fácil el aprendizaje de la música y a la vez lo haría mas divertido. Por ejemplo, MIDI provee una interactiva conexión entre un estudiante musical y una computadora, la computadora puede interpretar las acciones hechas por los estudiantes y proveer información acerca del resultado de la acción, esto acelera el aprendizaje del estudiante.

Es por esto, que el motivo de este proyecto es el de crear una herramienta para todo aquel interesado en crear, editar y mostrar sonidos digitales a través de instrumentos musicales compatibles con el protocolo MIDI, el cual captura y muestra las notas musicales de los instrumentos con fines de educación y aprendizaje.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Desarrollar una aplicación que capture las notas musicales desde cualquier instrumento compatible con las especificaciones MIDI para luego editarla, ejecutarla y guardarla en formato MIDI, el cual posea una interfaz gráfica amigable para el usuario y que las notas guardadas tenga el mismo tiempo real exacto con el que fueron tocadas las notas musicales.

1.2. OBJETIVOS ESPECIFICOS

- Asegurar la captura de notas musicales tocadas por el instrumento compatible de las especificaciones MIDI, para esto se creara un módulo que muestre todos los mensajes MIDI capturados con el instrumento MIDI con su significado.
- Eliminar los retrasos de tiempos cuando son capturadas las notas musicales, que tengan el mismo tiempo real como fueron tocadas las notas musicales del Instrumento musical.
- Diseñar una interfaz amigable que permita al usuario el manejo eficiente del aplicativo.
- Crear un módulo de edición de notas musicales y de notación solfa, para que el aplicativo se utilice con fines educativos para los maestros y con fines de aprendizaje para todo aquel interesado en música.
- Cotizar la implementación de un sistema de captura de notas musicales a través del formato MIDI.

2. ANTECEDENTES DEL TEMA O PROBLEMA

La música es la más antigua de las artes, dentro de ella existen una gran variedad de instrumentos musicales, al principio toda la música proviene de un sonido que nos llega al oído por medio de algo que vibra, llamado cuerpo sonoro (que puede ser un instrumento musical o no), existe algo que lo transmita que puede ser el aire, el agua o un medio sólido, el sonido entonces es producto de un cuerpo sonoro que vibra, la vibración que produce genera ondas de aire, que llegan a nuestro tímpano, no todos los sonidos son música, o no todo los cuerpo sonoros son instrumento musicales, hay dos grupos de sonido, el sonido musical y el ruido, la diferencia entre ellos es la armonía.

A través de los años la música como todas las cosas evolucionan, los instrumentos musicales ocupan varios medios para transmitir sus ondas musicales. En 1960 surgió la Música Electrónica, en el cual se creó el primer sistema dedicado a la música electrónica, conocido como Sintetizador, la música ocupa un nuevo medio para transportarse a través de impulsos de voltajes. En 1985 surge un estándar de comunicación entre los instrumentos electrónicos llamado MIDI (Interfaz Digital de Instrumento Musical). El mercado musical se transforma, fue todo un hit gracias a un gran número de fabricantes de equipos MIDI, hoy en día todo fabricante serio de la música incluye el estándar Midi en sus instrumentos y hasta la fecha es la mejor tecnología de comunicación electrónica de la música.

La tecnología Midi provee especial utilidad en la educación de la música, hay mucha teoría musical, en la que todos sabemos que es difícil de encontrar un método que enseñe y explique al mismo tiempo lo que estudiamos. Hay temas teóricos de forma aislada y al final queda una desconexión de lo que estudiamos con lo que tocamos. Con un sistema Midi les proveerá a los estudiantes de la música y a los profesores una gran herramienta para el aprendizaje de teoría musical, composición, técnicas de teclados musicales y muchas otras aspectos musicales, usando Midi los estudiantes

pueden integrar sus habilidades musicales tales como escuchar, leer, escribir, realizar y componer música con aspectos teóricos y conceptuales.

Existe poca información de Midi y una gran demanda de encontrar software de enseñanza musical, como una alternativa de solución a este problema, es el desarrollo de este proyecto, el cual permite que cualquier estudiante obtenga una información detallada para que logre capacitarse en las áreas de programación relacionada con la música, logrando obtener una herramienta de aprendizaje musical.

3. IMPORTANCIA Y JUSTIFICACIÓN DEL TEMA

Este proyecto surgió ante la necesidad de investigación de la tecnología MIDI y sus beneficios que posee a través de la música y el sonido como aprendizaje interactivo con el usuario, el cual posee un efecto de motivación con fines de educación.

Con MIDI convertimos la computadora en un estudio musical, un ejemplo específico en que un músico utilizaría Tecnología MIDI, es la posibilidad de conectar a la computadora un sintetizador y tocar la partitura, lo que se toque en ellos queda registrado como partitura en la computadora. Esto tiene una gran utilidad, ya que el músico puede tocar y luego escuchar lo que tocó, así como introducir arreglos y correcciones. Por lo tanto la computadora y el software de captura de notas musicales es una herramienta eficiente para la edición de sonido.

Un ejemplo como resultado de la creación de sonido o música, es la música que generalmente en modo de melodía acompaña los documentos o páginas de "Internet", el cual tiene un objetivo y hasta cierto punto se puede decir que ese objetivo es estético, pero se puede considerar imprescindible cuando se añade valor a la información en cuanto resulta mas fácil o agradable al usuario aprender lo que esta percibiendo, es por eso que la música digital capturado por el aplicativo de captura de notas musicales es muy importante, porque crea sonidos que interactúen con el usuario.

Al igual que ocurre con la imagen, los sonidos deben de ser adquiridos y luego manipulados de modo que tengan las características sonoras pretendidas para su implementación, La adquisición puede realizarse de varias maneras pero este proyecto pretende hacerla con la conexión de un instrumento musical que cuente con el estándar MIDI y que conecte a la interfaz MIDI de la tarjeta de sonido, en donde se interpretara la música con códigos de formato y normas especial, que es el protocolo MIDI.

El almacenamiento de sonido se puede realizar con el Estándar de archivo MIDI "SMF" que es el fichero que contiene una secuencia MIDI con su habitual "MIDI".

En resumen el proyecto pretende crear una herramienta aplicativa para todo aquel interesado en crear, editar y mostrar sonidos digitales a través de instrumentos musicales compatibles con el protocolo MIDI, el cual captura y muestra las notas musicales de los instrumentos con fines de educación y aprendizaje.

4. PROYECCIÓN SOCIAL

El proyecto esta enfocado para los maestros de música y para todo aquel interesado en aprender música y tecnología MIDI, este es el grupo de la población que es beneficiada por el proyecto, porque tiene muchos usos con fines de educación. Con MIDI logramos hacer muchas cosas como componer o editar música, aprender teoría musical, convertir la computadora en un estudio musical, aumentar la efectividad del software educativo, agregar sonidos a programas de entretenimiento educativos y habilitar el uso de la computadora para individuos discapacitados a través de la música.

5. DEFINICION DEL TEMA

Este Proyecto pretende el desarrollo de una aplicación que permita la captura de notas musicales desde un instrumento musical que cumpla con la especificación MIDI (Musical Instrument Digital Interface). Existen algunos problemas que deben ser resueltos tal y como la exactitud del tiempo de las notas musicales versus el tiempo real en que una nota es tocada. La aplicación deberá contar con una interfaz amigable para la edición de la pieza musical.

6. ENFOQUE CON QUE SE ABORDARA EL TEMA

Investigación y análisis de los conceptos, procesos, funcionamiento y configuración de todos los elementos involucrados de la tecnología MIDI, con el fin de desarrollar una herramienta gráfica, sencilla al usuario, modular(es decir se pueda ampliar y mejorar), que pueda simular todo estos elementos antes descritos y de esta manera permitir a los estudiantes de música una educación calificada a través de MIDI, basándose en los recursos y oportunidades actuales con los que se cuentan.

7. ALCANCES

1. El aplicativo mostrará cada mensaje que es capturado del instrumento musical MIDI y su significado, cada mensaje se representará en forma ordenada en virtud de su posición y tiempo a fin de poder reproducirlo o editarlo con posterioridad.
2. El tipo de notas musicales capturadas y almacenadas son de formato MIDI. Con su habitual extensión MIDI, se establecerá el estándar de archivo MIDI "SMF".
3. La implementación del aplicativo tendrá como enfoque la captura de las notas musicales en el tiempo real con el que fueron tocadas para luego almacenarla.
4. Posibilidad de escritura y lectura sobre unidades de almacenamiento MIDI.
5. Creación de tres editores que representan los eventos de forma decimal, hexadecimal y que visualice las notas en forma gráfica, para que sean fácilmente accesibles para su edición, los editores tendrán la opción de imprimir, el cual nos permite obtener copias de los resultados de cada editor.
6. Elaborar un estudio para la realización de un editor solfa donde se visualiza los pentagramas por pistas o Tracks.
7. Posibilidad de agregar pistas o track para capturar, reproducir, mostrar y guardar varias canciones MIDI independientemente de otras ya capturadas o guardadas.

8. Posibilidad de reproducir las notas musicales capturadas o almacenadas a través de la tarjeta de sonido de la computadora o a través de un instrumento MIDI.
9. Las opciones disponibles para los usuarios del aplicativo son :
 - Captura de notas musicales.
 - Abrir y guardar archivos en formato MIDI.
 - Visualización de los eventos Midi en formato decimal y hexadecimal.
 - Visualización grafica de las notas musicales de los eventos Midi.
 - Agregar Notas musicales al final de la primera Pista o Track.
 - Edición de la pieza musical a través de los eventos MIDI.
 - Reproducción de la secuencia MIDI.
 - Configuración del tiempo MIDI a traves del tempo (Beat por minuto), resolución (Tick por Beat) y interrupción del reloj (1 y 10 Microsegundos).

8. LIMITACIONES

1. El sistema funcionará únicamente sobre una plataforma Windows y no tendrá compatibilidad con otros sistemas operativos que no sean Windows.
2. El sistema es cien por ciento compatible con Windows 98, NT, ME, 2000 y XP.
3. El usuario solamente podrá introducir eventos MIDI a través de instrumentos compatibles con las especificaciones MIDI.
4. El sistema solo almacenará formato MIDI.
5. La tecnología MIDI soporta varios instrumentos físicos por medio de un cable multipuerto, pero en este proyecto solo se ocupará cable normal MIDI para un instrumento físico.
6. Visualización gráfica las notas musicales es de tipo melodía sin acompañamiento, se mostrará desde la nota "FA" de la segunda octava hasta el "MI" de la quinta octava y se reflejara la nota musical con un tiempo(símbolo de negra con clave de Sol).
7. La edición del aplicativo solo comprende las siguientes opciones:
 - Visualización de los eventos MIDI en formato Hexadecimal, decimal y en formato gráfico las notas musicales.
 - Agregar notas musicales al final de la primera pista o track.
 - No es capaz de cortar, copiar, deshacer y pegar los eventos MIDI en los 3 formatos de visualización.

CAPITULO II

ESTUDIO DE FACTIBILIDAD

1. FACTIBILIDAD TÉCNICA

Uso de la computadora en el aprendizaje de la musica.

La computadora es una excelente plataforma para el aprendizaje de la música, porque usan a plenitud su memoria interna, disco de almacenamiento y su video para la utilización de un software especial para el manejo de música a través de la tecnología MIDI. Ese software especial es llamado secuenciador. El secuenciador Midi es el área en el que realmente Midi se vuelve interesante, ya que hace posible grabar y tocar composiciones multiparte de Midi en cualquier estilo, desde techno hasta clásica. Un secuenciador es también llamado Grabadora multitrack de Midi, en el contexto general de grabado, el término "track" se refiere a grabar una parte musical de tal manera que pueda ser editada, borrada o regrabada independientemente de otras partes.

Una de las mejores razones de usar un secuenciador MIDI en una computadora es que no hay necesidad de contratar toda una orquesta o una banda de músicos para proveer todos los sonidos que nosotros necesitamos. En muchos casos el secuenciador Midi es mejor que el escribir manualmente los registros musicales, ya que con el secuenciador se hace posible tocar una parte exactamente como la primera vez que fue tocada.

El secuenciador MIDI por computadora es un programa didáctico en general, pero no pretende reemplazar al maestro de música, sino que está diseñada para proveer nuevas oportunidades de aprendizaje. Pero son ideales para aquel estudiante o maestro que desea profundizar en la teoría musical y en la tecnología digital de la música y son mejor utilizadas cuando son administradas por los estudiantes quienes

han dominado un conjunto de conocimientos y están listos para aplicar el conocimiento adquirido.

La notación solfa es una parte de la teoría musical que es demasiado grande y compleja de hacerla con el secuenciador, en este proyecto se estudió la posibilidad de hacerlo, se investigó a través de recolección de información tales como entrevistas con maestros de musica de instituciones como "CENAR", donde se comprueba que si es posible hacerla, pero estudiando minisiosamente las limitaciones que podría tener, porque que el solfa no hay que limitarlo debido a la dependencias que existen con muchas de las características que posee, incluso en la Institución de Música "CENAR" no han encontrado un software que realice notación solfa como debe de hacerlo, ya que cuenta con un software que realice solfa pero se limita en muchas de las características del todo el Solfa, el secuenciador de este proyecto no realiza la notación solfa, porque se limitó varias características tales como las tonalidades, las claves, la duración de los sonidos, fórmulas de los compases, tiempos fuertes y débiles, signos de repetición, signos de expresión, metrónomo, intervalos melodicos y armicos, acordes, arpeggios, campos armónicos, cadencias, resoluciones, entre muchos otros. Este proyecto no realiza notacion solfa, pero si tiene la capacidad de retomarlo posteriormente para realizarlo y hacer otro proyecto que solo se enfoque en hacer notación solfa, por lo grande y complejo que es.

Aspectos y Características.

Un Aspecto muy importante al usar un secuenciador Midi es su simulación de orquesta musical y el de plasmar la composición de la musica en formato teorico musical, esto se explica comparando un composición sin utilizar Midi y otra con la utilización de la tecnología Midi, el proceso de composición sin Midi empieza cuando el compositor se siente al piano, hace pruebas de sus ideas musicales, una vez que la tenga clara las escribe en un papel de manuscrito de varios sección de orquesta, el compositor imagina las partes que ya estan escritas mientras agrega más secciones, cuando termina todos los registro musicales estas pueden ser plasmadas en un escrutinio y cualquier alteración pude ser hecha, una vez terminada se contrata una orquesta para tocar la composición. Ahora utilizando Midi orquesta puede se compuesta por un teclado sintetizador, el cual simula todos los instrumentos necesarios para tocar una pieza musical, el sintetizador esta conectado a un secuenciador, en lugar de escribir los registro de la música en un manuscrito, el compositor podria grabar secciones de musica dentro del secuenciador, en vez de imaginarse las partes que ya estan escritas, el secunciador es capaz de tocar de vuelta lo grabado y escucharlo exactamente como lo grabo y editandola fácilmente con la interactividad de la computadora, todo esto hace que el secuenciador sea más práctico y fácil para componer música, sin consumir demasiado tiempo.

Estudio para la elección del entorno visual de desarrollo a utilizar para la construcción del secuenciador

Existe una gran cantidad de entornos de desarrollo de los cuales un programador de aplicaciones puede elegir, ya sea en el desarrollo de aplicaciones para Windows, Linux y otros sistemas operativos.

Estos entornos utilizan lenguajes para realizar tareas que el programador especifique. Para el lenguaje Basic existe Visual Basic, Realizer, GFA Basic; para C están Watcom C, Symantec C, Visual C++, Borland C, Zortech C. Para Pascal hay poco soporte. Por muchos años existió Microsoft Pascal, Turbo Pascal, FPK Pascal, Borland Pascal, Delphi y por ultimo Java el lenguaje multiplataforma.

Debido a que el software de secuenciador utilizará drivers de puerto MIDI y presentará elementos graficos de notas musicales, será necesario el uso de una lenguaje de bajo y alto nivel, la herramienta tiene que ser visual para que permita representación de objetos gráficos y que además permita personalización del comportamiento y sus características de los objetos. Por lo que la elección del software a utilizar no solo depende del lenguaje sino que también, que ofrezca un entorno en el que se puedan crear ventanas y mensajes de Windows en las que se puedan desplegar los elementos representativos.

Este estudio se centra en 2 preguntas importantes para la elección de ese entorno visual de desarrollo a utilizar para la construcción del secuenciador, Los aspectos son los siguientes:

¿Qué se esta tratando de hacer?

¿Qué clase de características se desean implementar?

Estas preguntas se contestaran en las conclusiones del estudio, luego de mostrar las características de los entornos de programación comercialmente existentes.

Herramientas de programación visual.

Las herramientas de programación visual comparten casi una misma apariencia básica: un formulario, una caja de herramientas de componentes, una ventana de propiedades y una ventana de código.

VISUAL BASIC

Visual Basic es un entorno de desarrollo de aplicaciones para Windows con muchas herramientas útiles, sin mencionar el total acceso al entorno de Windows, por lo tanto un programa escrito en VB puede hacer casi todo lo que hace Windows. Además ofrece un conjunto de características orientadas al desarrollo de aplicaciones personalizadas de una forma fácil y rápida.

El lenguaje utilizado en Visual Basic es directo, con la revisión de sintaxis usando colores y con "compilador real" (sin usar el P-code) por lo que la ejecución de un programa es bastante razonable. Además de mantener el lenguaje BASIC fácil-de-aprender al que se le ha añadido la funcionalidad de Windows.

Un programa generado por Visual Basic está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interface gráfica. Es por tanto un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos.

La mayoría de las características nuevas de Visual Basic están orientadas más que todo al acceso a datos y de hacer a Visual Basic una opción viable como ambiente de desarrollo de aplicaciones integradas complejas. Muchas de las nuevas características son respuesta a las quejas comunes de desarrolladores propuestas en grupos de noticias y revistas.

Una de las razones de la popularidad de Visual Basic es el modelo de componente del que es pionero en la creación de pequeñas y medianas aplicaciones, ofreciendo

mejoras en la productividad y la facilidad de uso. Además, Visual Basic maneja de mejor manera el ambiente gráfico.

Características.

Programa con el más popular lenguaje de programación para crear rápidamente aplicaciones y componentes de alto rendimiento en un entorno RAD.

Construir aplicaciones y componentes de código nativo que usa la misma tecnología de compilador de clase mundial como en Visual C++. Las aplicaciones pueden ser optimizadas en velocidad y tamaño, y en muchas otras formas, mejorar aun más la ejecución.

Incrementa la escritura de programas y la productividad con la tecnología IntelliSense. Sintaxis al instante y terminación de sentencias usando características como Complete Word (completar palabra), Quick Info (Información rápida) y DataTips.

Creación fácil de un amplio rango de componentes re-usables que encapsulan propiedades o datos legales, incluyendo controles ActiveX.

Un entorno RAD muy intuitivo que promueve la productividad y ayuda a los desarrolladores a crear y desarrollar rápidamente aplicaciones cliente/servidor.

Creación de programas para Internet usando herramientas y técnicas de programación de Visual Basic, que pueden ser accedidas desde cualquier navegador en cualquier plataforma.

Incorpora el entorno de programación de datos mejorado y proveedores OLE DB. ADO (ActiveX Data Objects) para acceder a datos a través de la LAN o Internet.

Uso del Microsoft Data Engine como parte de las aplicaciones y obtener una compatibilidad completa con grandes bases de datos en SQL Server.

Construcción de aplicaciones con auto-reparación y bajos costos de distribución con el nuevo Visual Studio Installer.

VISUAL C++

C++ es un supeconjunto de lenguaje C, conserva todas las características C incluyendo su eficacia y flexibilidad a la hora de trabajar con la interfaz del hardware y software, su programación del sistema a bajo nivel y su eficiencia, economía y expresiones eficaces. Sin Embargo, C++ introduce al lenguaje C en el mundo dinamico de la programación orientada a objetos y le convierte en una plataforma para la abstracción de problemas de alto nivel, donde se puede ir más alla que ADA en este aspecto a Modula-2, a la vez que conserva la eficacia de ejecución concisión de C.

Este lenguaje hibrido combina las construcciones del lenguaje de procedimiento estandar, y el modelo orientado a objetos, que puede aprovechar completamente para crear una solucion exclusivamente orientada a objetos para un problema.

Desde la primera version de Windows, los programadores han escrito aplicaciones de windows mediante la API (Interfaz de programacion de aplicaciones) de windows atravez de las funciones de C, a las que se llaman para interactuar con Windows, practicamente la API de windows fue fundamentalmente escrita para los programadores de C, lo que implica que el Sistema Operativa de microsoft esta bastante relacionado con el lenguaje C.

El problema de C es que es un lenguaje estructurado y para un mejor empaquetado de codigo de programacion, se necesita una interfaz que interactuaba con el sistema operativo con un lenguaje orientado a Obejtos, para eso Microsoft creo la biblioteca de clases MFC, el cual encapsula código que el programador no necesita saber para el funcionamiento de sus aplicaciones en Windows.

Otros lenguajes como Visual Basic necesita realizar una gran cantidad de trabajo sucio para actuar como interfaz para la API de Windows, aunque hoy en dia con la Biblioteca de clases de .Net Framework proporciona un conjunto de clases que se

podrán utilizar en cualquier lenguaje .NET para interactuar con el Sistema Operativo, entre ellos Visual Basic y inclusive Visual C ++.

En resumen no existe un lenguaje más eficaz y flexible que C++ para realizar aplicaciones en Windows que interactúan con el Software y Hardware de la computadora, ya que es un lenguaje de bajo y alto nivel.

Características.

Visual C++ contiene un entorno de desarrollo de aplicaciones para Windows con muchas características, tales como la interfaz de dispositivo gráfico de Windows llamada la GDI, programación basada en recursos, la interfaz de programación de aplicación (API) win32, Bibliotecas de enlace dinámico (DLL's), el estándar de aplicaciones de windows MFC, Biblioteca de plantillas activas de Microsoft(ATL's) para construir y utilizar controles ActiveX y por último la versión .net que es una biblioteca de clases orientadas a objetos para proporcionar todas las herramientas necesarias para escribir una amplia gama de programas . Por lo tanto un programa escrito en VC++ puede hacer todo lo que hace Windows y mucho más. Además ofrece el desarrollo de aplicaciones orientadas a las interfaces físicas de la computadora.

¿Visual C++ o Visual Basic?

Interfaz del entorno

Tanto Visual Basic como Visual C++ cuentan con entornos de desarrollo basados en una interfaz de última generación, con múltiples ventanas acoplables entre sí, barras de botones configurables, conjuntos de componentes agrupados en páginas, menús emergentes sensibles al contexto y un impresionante número de opciones correctamente dispuestas en los correspondientes menús desplegados.

Formulario

En principio Visual Basic usa un modelo MDI, de tal forma que las ventanas correspondientes a diseñadores y edición de código están contenidas en una ventana principal. Visual C++, usa tanto el modelo SDI como el MDI que le es habitual desde el nacimiento con la Api de windows en C, despues con los MFC y por ultimo con las .NET Framework. Los usuarios de Visual C++ y Visual Basic podrán elegir el modelo que quieren, ya que es posible cambiar mediante una opción de MDI a SDI. Aunque hay un mejor control de la interfaz a través de C++, ya que hay varias alternativas de utilizar interfaz gráfica, tales como las del uso de la herramienta de recursos para utilizarlos en MFC o en ATL y la de las clases de la biblioteca de Windows Forms del .NET (teniendo en cuenta que en Visual Basic .NET también se pueden utilizar).

En Visual C++ el desarrollo del entorno gráfico es más flexible porque es a través de la definición y aplicación de clases heredadas según su clase base gráfica, en cambio en Visual Basic es código ya establecido y que no es necesario conocer el código, si no que lo encapsula y solo se puede cambiar a través de sus propiedades de formulario a nivel gráfico.

Componentes

Uno de los apartados más importantes en una herramienta de desarrollo RAD es, sin duda, el número y la calidad de los componentes que incorpora. De ello depende que el usuario tenga más o menos posibilidades y que, en ocasiones, se vea en la necesidad o no de buscar componentes de terceros que satisfagan sus necesidades. También es importante conocer el modelo o modelos de componentes para los cuales está preparada dicha herramienta, ya que ello determinará la cantidad de componentes que podremos encontrar en el mercado.

La primera gran compatibilidad entre Visual C++ y Visual Basic la encontramos, precisamente, en los modelos de componentes que utilizan. Al ser un producto de Microsoft, Visual Basic como es lógico usa controles ActiveX, el modelo *oficial* de la plataforma Windows.

Lenguaje

El lenguaje usado por Visual Basic es BASIC (*Beginner's All-purpose Symbolic Instruction Code*), un lenguaje desarrollado en la década de los sesenta por dos profesores estadounidenses, Kemeny y Kurtz, como una versión simplificada de FORTRAN (*FORmula TRANslator*), que fuese más accesible para los alumnos principiantes en computación.

El lenguaje C++ tiene su origen con el Lenguaje C. Aunque C++ se asemeja a BCPL y ALGOL 60, también contiene componentes de Simula 67, C++ representa una evolución y mejora de las características del lenguaje C, a Bjarne Stroustrup, de Bell Labs, se le atribuye el desarrollo del lenguaje C++ a comienzos de 1980, C++ se desarrolló originalmente para solucionar algunas simulaciones controladas por eventos muy rigurosos en los que las consideraciones de eficacia excluían el uso de otros lenguajes. Uno de los objetivos claves de C++

era mantener la compatibilidad de C y la gran mejora o la más importante es su compatibilidad con la programación orientada a los objetos.

Visual Basic es un lenguaje sencillo de aprender y, en general, bien estructurado, aunque en él aún permanecen elementos del pasado, como `On Error Goto`, que deberían haber sido sustituidos por estructuras más actuales y flexibles. A pesar de que es posible crear módulos de clases y manipular objetos de modo polimórfico, este mecanismo no se basa en un lenguaje totalmente orientado a objetos sino en el uso de interfaces COM. Es posible obtener direcciones de procedimientos, con el fin de usar llamadas *callback*, pero el manejo de punteros es oscuro e indocumentado por parte del fabricante, lo que redundaría en un mayor esfuerzo por parte del desarrollador. La integración de Visual Basic con la tecnología ActiveX es sobresaliente. Puede utilizarse cualquier objeto que soporte automatización y, al tiempo, cualquier objeto creado con Visual Basic se convierte ya en un objeto ActiveX. Para usar objetos COM simples, no automatizables, es necesario disponer de las librerías de tipos correspondientes, no existiendo modo alguno de crearlas desde el propio entorno de Visual Basic.

El lenguaje de programación más importante de Microsoft para las aplicaciones de 32 bit es C++, anteriormente, el lenguaje ensamblador tenía una función más importante en situaciones donde la velocidad es primordial, aunque se puede comprobar que la mayor parte de código de Windows se escribe en C++. Microsoft ha proporcionado todas las herramientas necesarias con el compilador para desarrollar programas en Windows de 32 bits desde el entorno del lenguaje. En la actualidad es el lenguaje más utilizado e implementado por una gran cantidad de

fabricantes, ya que C++ es el lenguaje más eficaz y flexible para interactuar con el software y el hardware de la computadora.

Depuración.

Difícilmente una aplicación llega a su estado definitivo sin pasar previamente por un proceso de depuración, ya que no todos los errores que se introducen pueden detectarse durante la compilación.

Ambas herramientas disponen de las habituales funciones de ejecución paso a paso, incluyendo las opciones de ejecución de procedimientos completos o instrucción a instrucción. Disponer o anular un punto de parada se efectúa de la misma forma: simplemente pulsando con el botón izquierdo del ratón en el margen izquierdo del editor de código, a la altura de la línea deseada. No obstante, en Visual C++ existe la posibilidad de crear puntos de parada que actúan al cargarse un determinado módulo o alterarse un cierto dato.

Características exclusivas.

Una característica exclusiva de Visual Basic es la posibilidad de crear aplicaciones DHTML, consistentes en un documento HTML dinámico y una librería de código Visual Basic que es ejecutado desde dicho documento. De esta forma es posible crear documentos HTML dinámicos sin exponer el código, aunque su uso queda restringido a los usuarios de Internet Explorer. Es una opción muy válida sobre todo en *intranets*, donde el administrador tiene un total control sobre el *software* que utilizan los clientes.

En el haber de Visual C++ encontramos las opciones de creación de librerías de enlace dinámico con funciones y objetos, desarrollo de servicios NT, creación de aplicaciones multihilo sin necesidad de usar ejecutables ActiveX, creación de objetos COM automatizables y no automatizables y el desarrollo de objetos y módulos CORBA con servicios que pueden compartirse con otros lenguajes, como Java y

PowerBuilder. La inclusión de componentes de toma de decisiones y las amplias posibilidades de los componentes Internet también se han de tener en consideración, así como la facilidad para crear aplicaciones en múltiples idiomas.

Conclusión.

No hay duda que tanto Visual C++ como Visual Basic son magníficas herramientas de desarrollo de aplicaciones para Windows, Pero no compiten prácticamente al mismo nivel en la mayoría de los aspectos tratados anteriormente, hay muchos aspectos en los que Visual C++ supera a Visual Basic.

Pero también, existen ciertos factores que tienen prioridad que nos hacen elegir a Visual C++ como la herramienta de programación para este proyecto.

Retomando las preguntas hechas al inicio de este análisis, nos disponemos a contestarlas con lo expuesto anteriormente.

¿Qué se esta tratando de hacer?

Primeramente, lo que pretende este proyecto es hacer una aplicación de captura de notas musicales en formato MIDI, en el cual tiene que ser un lenguaje de bajo nivel que interactue con el driver de la interfaz MIDI y un lenguaje de alto nivel que interactue con el usuario a nivel gráfico.

Visual Basic viene equipado con una cantidad increíble de controles como ya se ha mencionado, pero el hecho es que los controles incorporados en Visual Basic no son suficientes funcionales para la utilización de todos las características necesarias para el manejo del aplicativo MIDI, como grabar, tocar, reproducir y escribir en disco la captura de nota musical en formato MIDI.

¿Qué clase de características se desean implementar?

Visual basic es una herramienta RAD, que debido a sus capacidades, esta más orientada para la construcción de aplicaciones empresariales que necesitan manejar grandes cantidades de datos con el máximo rendimiento posible.

Una característica muy útil de mencionar de Visual C++ es la rapidez, la eficiencia y la flexibilidad de llamar a cualquier funcion Win32 en cualquier momento.

2. FACTIBILIDAD ECONÓMICA

Costo de comprar hardware y software propietario de secuenciador MIDI.

Costo de comprar hardware y software propietario de secuenciador MIDI

Cant	Dispositivo	Descripcion	Unitario	Total
1	Computadora	PC Clon	\$1,000	\$1,000
1	Cable Midi	Cable Midi	\$40	\$40
1	Sintetizador	Sintetizador Casio CT-700	\$240	\$240
1	Sistema Operativo	Windows XP Pro	\$150	\$150
1	Impresora	Cannon BJC 2000	\$40	\$40
1	Secuenciador	Cubase Sx/Sb	\$1000	\$1000
			SubTotal	\$ 2470
			IVA 13%	\$ 321
			Total	\$ 2791

Costo de comprar hardware y de implementar el Aplicativo de captura de notas musicales en formato MIDI

Costo de implementacion de secuenciador MIDI

Cant	Dispositivo	Descripcion	Unitario	Total
1	Computadora	PC Clon	\$1,000	\$1,000
1	Cable Midi	Cable Midi	\$40	\$40
1	Sintetizador	Sintetizador Casio CT-700	\$240	\$240
1	Sistema Operativo	Windows XP Pro	\$150	\$150
1	Impresora	Cannon BJC 2000	\$40	\$40
1	Secuenciador	Secuenciador Midi	\$300	\$300
			SubTotal	\$1770
			IVA 13%	\$230,1
			Total	\$ 2000,1

Costo del proyecto

Papelería	\$ 110.00
Impresiones (Cartuchos de Tinta)	\$ 120.00
Internet	\$ 160.00 (\$ 20.00 mensuales)
Electricidad	\$ 126.25 (\$ 25.25 mensuales)
Varios	\$ 100.00
Horas hombres	\$ 1680.00 (\$10x8horasx21dias)

Total \$ 2296,25

Nota: Los costos de electricidad e Internet son estimados que se basan en los recibos mensuales de la residencia donde se desarrolló el proyecto.

3. FACTIBILIDAD OPERATIVA

El Secuenciador para la institución, el profesor y el estudiante de musica, será una aplicación Windows completamente amigable y ejecutable en cualquier Sistema Operativo Windows (9x, Me, NT, 2000 y XP).

Su uso esta designado específicamente para los estudiantes de Música y toda persona interesada en aprender música a través de MIDI.

Para su instalación los archivos de instalación estarán en un CD con un archivo ejecutable (setup.exe), este cargará el asistente el cual instalará en el disco duro local todos los archivos, librerías y componentes necesarios para el aplicativo Midi. Luego de instalado, el usuario será capaz de cargar el secuenciador y comenzar a capturar notas musicales para luego editarlas.

El usuario trabajará bajo un ambiente gráfico, de ventanas, menús y formularios. Para poder capturar las notas musicales el usuario dispondrá de una pantalla de captura donde se mostrará los mensajes midi capturados y grabados. Después de capturar los mensajes midi, el secuenciador tiene varias opciones que estarán disponibles.

Una de las opciones es reproducir lo grabado en el dispositivo interno o externo, el dispositivo interno que es el sintetizador que trae la tarjeta de sonido y el dispositivo externo es el equipo MIDI en el que se tocaron las notas musicales que son grabadas por el secuenciador MIDI, también otra opción es guardar en disco lo grabado para luego reproducirlo o editarlo.

El secuenciador contará con un módulo edición de notas musicales para cambiar, agregar y borrar notas musicales. Este modulo consiste en editores de lista de eventos midi en formato Hexadecimal, decimal y mostrara gráficamente las notas musicales, donde se explica cada mensaje MIDI por descripciones específicos de los conocimientos musicales, luego de editarlos tendra la opciones de guardar los cambios o imprimirlos en papel las notas musicales para que el estudiante o usuario pueda ver el resultado de su composición.

Sin secuenciador MIDI.

Ventajas

- Explicación más completa de Teoría Musical con profesor de música.

Desventajas

- Costo elevadísimo para aprendizaje Música y Teoría Musical.
- Dificultad y desmotivación en el aprendizaje de la Musica.
- No tocar las notas musicales exactamente como la última vez.
- Subutilización del funcionamiento del equipo Musical.
- Costo elevadísimo en montar un estudio musical.
- Varios equipos para tocar una melodía con varios sonidos.

Secuenciador MIDI.

Ventajas

- Grabar todo la composición en disco duro, para luego usarla o editarla.
- Costo bajo en aprendizaje de la Música y Teoría Musical.
- Costo bajo en tener un estudio musical.
- Diversión interactiva y facilidad aprender notas musicales a través del secuenciador MIDI.
- Explorar y aprovechamiento de equipo musical.
- Utilizar las notas grabadas para motivos educativos o comerciales a través de la computadora.
- Mezclar notas guardadas con notas tocadas en tiempo real.
- Editor notas musicales a nivel de comando Midi.
- Exploracion del mundo MIDI, para experimentar con una gran variedad de opciones que faciilita y expande el mundo musical.

Desventajas

- El secuenciador nunca sustituirá un profesor de música.

CAPITULO III

METODOLOGÍA DE LA INVESTIGACION

1. METODOLOGIA DE LA INVESTIGACIÓN

En este caso como el tema se basa en la música, el desarrollo de este proyecto involucra una institución referente al tema, dicha institución ofrecerá un ambiente apropiado para conocer el mundo del aprendizaje de teoría musical. El papel de la institución es muy importante en el proyecto, porque se involucrara directamente con el proyecto a nivel de investigación y prueba de producto final, esta institución tendrá que evaluar el aplicativo y actuara como un ente crítico para la segunda fase del proyecto y presentación de la segunda defensa.

Las etapas que pueden distinguirse en el desarrollo de la metodología de la investigación son:

1. Análisis de la solución: Que es la primera Fase del modelo de desarrollo de software, debido que se ocupará el modelo lineal secuencial como elementos claves para los procesos de desarrollo del aplicativo, el cual consiste en cuatro fases que son:

- Análisis de la solución.
- Diseño de la solución.
- Código o desarrollo de la solución.
- Pruebas y depuración.

2. Diseño de la solución: una vez comprendidos los factores que participan en la captura de notas musicales, se está preparado para diseñar la solución del problema planteado. Las tareas que involucra el diseño de la solución son:

- Captura de notas musicales.
- Captura de notas musicales en tiempo real como fueron tocadas.
- Diseño de interfaz de usuario.

- 3. Desarrollo de la solución:** consiste en desarrollo del código de todos los elementos definidos en la etapa de diseño de la solución.
- 4. Prueba y depuración:** esta etapa consiste en corregir las diferencias que puedan surgir entre lo diseñado y lo implementado, es decir, esta fase busca que los objetivos iniciales del proyecto sean cumplidos.
- 5. Análisis de resultados:** que es recolectar toda la información y desarrollo que se realizaron con las actividades anteriores para luego analizar lo que se ha logrado.

2. TECNICAS A UTILIZAR EN LA INVESTIGACIÓN

En esta fase se utiliza técnicas de ingeniería para el desarrollo de la investigación, en el cual se involucra la institución musical, las técnicas que se utilizaron son:

- **Entrevistas:** Esta herramienta consiste básicamente en reunirse con una o varias personas, a mencionar, entendidos como instituciones involucradas en música, cuestionarlos en forma adecuada para obtener información. Este medio es el más usado y el que brinda información más completa y precisa, puesto que el entrevistador, al tener contacto con el entrevistado, además de obtener respuestas, puede percibir actitudes y recibir comentarios.
- **Métodos y Técnicas de Investigación Documental:** Esta técnica consiste en seleccionar y analizar aquellos escritos que contienen datos de interés relacionados con el estudio y toda aquella documentación que pueda aportar información relevante a la investigación.
- **Evaluación de Proyectos:** La evaluación de proyectos es una actividad inherente al desarrollo y elaboración de un estudio ya que en ella intervienen disciplinas como la estadística, investigación de mercado, investigación de operaciones, ingeniería de proyectos y evaluación financiera. Por lo tanto, es un soporte para las técnicas ingenierías utilizadas, durante la investigación y análisis del proyecto.

3. INVESTIGACION PRELIMINAR Y RECOLECCION DE INFORMACIÓN.

La finalidad de una institución o academia de música es la enseñanza de música y como toda academia musical cuenta con una gran variedad de instrumentos musicales tales como la guitarra acústica, guitarra eléctrica, piano, teclados sintetizador, Violín, Flauta y muchos más. Toda clase de Música dependiendo del instrumento consta de una clase teórica y una práctica.

Estudio del objetivo del proyecto.

El estudio se basó en investigar teoría musical para aplicarla en el secuenciador MIDI, en la que se entrevistó al Lic. Carlos Perez encargado de enseñanza musical de la institución CENAR, en el cual se recolectó información necesaria para desarrollar el aplicativo y establecer los alcances y limitaciones del proyecto.

CAPITULO IV

DESCRIPCIÓN DEL PROYECTO

1. SITUACION ACTUAL

La mayoría de academias no ocupan software musical para capturar notas musicales para fines educativos, para ocupar el software tendrán que adquirir computadoras para implementar el software del aplicativo de captura, así poder implementarlo con el objetivo de practicar la música de una manera fácil e interactiva con el uso del conocimiento Teorico aplicado.

2. RESULTADOS ESPERADOS

Los resultados que se pretenden alcanzar con el desarrollo del aplicativo de captura de notas musicales son:

- Facilitar el aprendizaje de notas musicales a los estudiantes a través de una herramienta grafica, amigable e interactiva por medio de la cual realicen sus prácticas.
- Dar la oportunidad a los estudiantes de música, de poder practicar los conceptos teoricos de música.
- El usuario podrá simular una orquesta musicales a través del aplicativo sin necesidad de tener varios instrumentos tocando al mismo tiempo.
- Mejorar la calidad de enseñanza de los maestros de música y el desarrollo de habilidades en el manejo de instrumentos musicales.
- Realizar un secuenciador que tenga la capacidad de expansión para futuras mejoras o requerimientos.

3. DIFICULTADES TECNICAS ENCONTRADAS

Las dificultades encontradas para la realización de Notación Solfa son:

- Se Necesita conocer perfectamente toda la teoría musical de la notación solfa a través de un estudio profesional de academias acreditadas a realizarlo tales como "CENAR".
- No hay que limitar la notación solfa, debido a las muchas dependencias que existen en ella, no es profesionalmente aprobado presentar la notación solfa con limitantes en el aplicativo.
- Se necesita crear un algoritmo que administre a la perfección la visualización gráfica de las notas Musicales en base a los tiempos de los eventos MIDI capturados por el secuenciador y que este dispuesto a cambiar según que el tipo de fórmula de compas Musical del pentagrama, esto es clave para la realización de la notación solfa en el aplicativo y es suficientemente complejo para realizarlo en esta tesis.

CAPITULO V

MARCO TEORICO

1. MARCO HISTORICO

MIDI

El estándar de Interfaz Digital de Instrumentos Musicales conocido como MIDI fue creada alrededor del año 1985. Originalmente fue concebido por un consorcio de fabricantes de sintetizadores tales como Roland, Yamaha, Oberheim y Sequential Circuits, con el objetivo de conectar juntos los instrumentos musicales, esto fue desarrollado dentro de un protocolo que escoge el control de todo desde el sonido hasta generadores de escenarios de luces.

MIDI es una especificación para un protocolo principal de comunicaciones usado para control de un instrumento musical a través de un conjunto de comandos o de lenguajes midi, detalla la manera electrónica de cómo estos comandos son transmitidos y recibidos.

Las especificaciones MIDI es presentado en un documento titulado "MIDI 1.0 detalle de especificación", llamado también como "the spec" el cual es publicado y distribuido por la asociación Internacional de MIDI.

El original de MIDI y todavía primario es usado en interpretaciones y composiciones musicales. En los años recientes esto se ha movido en diferentes áreas, tales como editor de audio y producción, escenario de luces y otros aspectos de interpretaciones en vivo. Estos grupos de comandos describen el lenguaje que es diseñado específicamente para transmitir información acerca de funcionamiento musical, esto quiere decir que el protocolo no transmite música, si no como este comando debe ser interpretado por el instrumento MIDI.

Los sintetizadores MIDI inmediatamente fue un hit en el mercado desde que fue publicado en 1985 el oficial documento de las especificaciones MIDI, desde ese año cualquier instrumento electrónico serio incluye las capacidades MIDI.

MIDI se baso en dos principales razones: universalidad y expandibilidad. Cada fabricante de Equipo MIDI que usa el término de MIDI en cualquiera lado de su equipo o en su documentación se espera que implente la especificación correctamente. Esto se refiere que la unidad puede aceptar y actuar correctamente los datos generados por otro equipo MIDI y que cualquier dato MIDI generado por el debe ser entendido por otro cualquier equipo MIDI. Esto significa que cada comando MIDI tiene el mismo significado que el recibo como el que transmite.

Las especificaciones MIDI fueron deliberadamente diseñado con fallas en sus comandos que no estaban todavía definidos, hasta que desde 1987 fueron agregados y desde entonces no hay posibles problemas, todavía hoy en día la especificación MIDI es referida como "1.0", aunque este ha sido expandida, esto es porque comandos originales todavía existen en la especificaciones.

SECUENCIADORES

Mucho antes de que se desarrollara la tecnología MIDI, ya existían aparatos que trabajaban en conjunto con los viejos sintetizadores análogos, como el Moog, ARP, SYSTEM y todos esos cacharros que revolviaron al mundo desde la década de los 50. Estos aparatos que trabajaban desde entonces, eran los secuenciadores.

Un secuenciador, es un equipo electrónico que en cierto ciclo repite una serie de notas, pulsos, timbres o ruidos.

Como era cíclico, se podían dar perfectamente bloques que cuadraran en compases de 2, 4, 8 o lo que fuera musicalmente aceptado. El rock progresivo encontró estos equipos como una gran ayuda para el desarrollo de su música. A mediados de los 70, aparece un secuenciador, el Roland TR-707, que inició el camino para que después se integrara esta tecnología a MIDI. Este aparato lograba tener secuencias de arpeggios (a lo que se le llamaba arpegiador), de líneas de bajo y opción de crear percusiones verdaderamente impresionantes. Esto lo hacía mediante métodos analógicos.

Usando este principio como modelo, aparecen los primeros secuenciadores, que tienen la capacidad de almacenar algunas notas en memoria, y repetirlas en el canal MIDI. Esto era muy limitado, pues había poco desarrollo en el campo de la "memoria"

de las máquinas, así que no se podía pensar en ese momento que fuesen la solución de ningún problema, sino daban la oportunidad de hacer sonar esos equipos que ya contaban con opciones multitímbricas y multicanales.

Así que el primer gran salto, lo dan empresas como Roland, Yamaha y Korg, que aparecen, simultáneamente equipos que hacen secuencias, y que tienen capacidad MIDI, y también sintetizadores, con sus propios secuenciadores de 8 canales, para poder diseñar las primeras piezas completamente construidas mediante el sistema MIDI.

Sin embargo, nuevamente aparece la falta de recursos tecnológicos, pues no hay mucho espacio dónde guardar la información de una secuencia, y entonces, el mundo del MIDI se dirige a una solución muy práctica: Usar a las computadoras, como medio de almacenaje (por su mayor capacidad de memoria) y como secuenciadores (por su alta resolución matemática).

Desembocamos a este tema, pues sin que la computadora sea un instrumento musical, pronto se volvió el corazón de los sistemas MIDI, con sofisticados secuenciadores, como Performer de Mark of the Unicorn, Vision de Opcode o Master tracks de Passport. Y entonces, aparece uno de los últimos conceptos que se deben aprender para dominar las bases de MIDI. Ya que MIDI necesita que exista una conexión entre un instrumento y otro, y las computadoras (a excepción de Atari, que fue diseñada desde un principio con puertos MIDI out e in) no cuentan con este sistema desde un principio, se agregan a estos sistemas lo que llamamos la Interfase MIDI. Una interfase MIDI es un aparato que se conecta tanto a la computadora como a los sintetizadores.

Obviamente necesita contar, tanto con las entradas DIN, como con una conexión que exista de antemano en la computadora. Por ello, las interfases MIDI para computadoras Macintosh vienen con un puerto de comunicaciones, y las que son para PC y Amiga, vienen ya sea con una entrada al puerto paralelo, o son internas, es decir, se pueden agregar a uno de los slots de la tarjeta madre del CPU.

Pero no olvidemos que una computadora es mucho mas versátil y potente que un simple instrumento musical, así que gracias a estas máquinas, obtenemos 2 ventajas nuevas para manejar estos sistemas:

- 1) El uso de gran cantidad de Canales de secuencia. (lo que llamaremos Tracks)
- 2) La adopción de Puertos de envío de canales MIDI.

El primer punto, se debe principalmente a que la gran capacidad de memoria le permite a la computadora manejar fácilmente una gran cantidad de pistas. Para la máquina, no son sino datos que entran y salen, así que los maneja con gran fluidez y precisión. Así que hasta en los secuenciadores mas simples, encontramos por lo menos 64 pistas o Tracks de grabación. Pero la existencia de esta gran cantidad de tracks, genera nuevamente otra perspectiva.

TARJETA DE SONIDO Y MIDI

El sonido digital siempre ha tenido diversos formatos. El sonido en formato digital tiene un problema, y es su excesivo espacio para almacenar relativamente poca información. Se pueden hacer los cálculos fácilmente: audio a 44,1KHz, con 16 bits y en estéreo, nos da 172 Kb/segundo (10,3 MB por minuto). Este método de almacenar el audio digital "tal cual" es el utilizado en los ficheros .wav o en el Cd-audio. Sin embargo, no resulta útil para los profesionales del sector (sobre todo para los compositores); imaginar la cantidad de disco duro y, sobre todo, memoria que son necesarios para trabajar a pleno rendimiento con el audio digital. ¿Cuál es la posible solución? El formato MIDI (Musical Instrument Data Interface) Al contrario que el audio digital, el formato MIDI no es el sonido grabado, sino principalmente las notas musicales que lo componen. Cualquier fichero MIDI ocupará poquísimos espacio, debido a que tan solo es necesario almacenar las notas que están sonando en cada momento. El formato MIDI nació para estandarizar el comportamiento de los distintos instrumentos digitales, para que las mismas notas sonaran "igual" en los distintos instrumentos. Hoy en día existen teclados MIDI (los sintetizadores), pianos MIDI, violines MIDI, flautas MIDI, baterías MIDI, e incluso gaitas MIDI.

Pues bien, en el caso del ordenador, alguien tendrá que encargarse de reproducir las composiciones MIDI. Y por supuesto, la solución está en aprovechar nuestra tarjeta de sonido. Como el formato MIDI no son más que notas, tendremos que obtener los sonidos de algún sitio, y existen dos opciones. La síntesis FM es la más económica. Hasta el momento, y desde hace mucho, ha sido la solución más empleada.

La síntesis FM no es más que un pequeño procesador que se encarga de imitar el sonido mediante el empleo de fórmulas matemáticas trigonométricas. Y en cierto modo, da mejores resultados de los esperables.

La solución que aportan la mayoría de tarjetas domésticas (desde la SoundBlaster AWE 32 y la Gravis UltraSound), es la inclusión de la síntesis por Tabla de Ondas (WaveTable). Esto no es más que el tener los Sonidos de los instrumentos grabados (a partir de instrumentos reales) en una memoria incluida en la propia tarjeta (ROM que normalmente se puede ampliar con RAM para añadir nuevos y mejores sonidos) o utilizando la memoria del ordenador, en cuyo caso deberá tener conector PCI en lugar de ISA. Con esto conseguimos una calidad mucho mayor en la reproducción de canciones MIDI. Actualmente sólo la Crystal, SoundBlaster 16 y las compatibles Yamaha OPL3. (así como algunas soluciones integradas en placas base y otras de fabricante desconocido) funcionan sin tabla de ondas.

2. MARCO CONCEPTUAL

Como Trabaja el MIDI

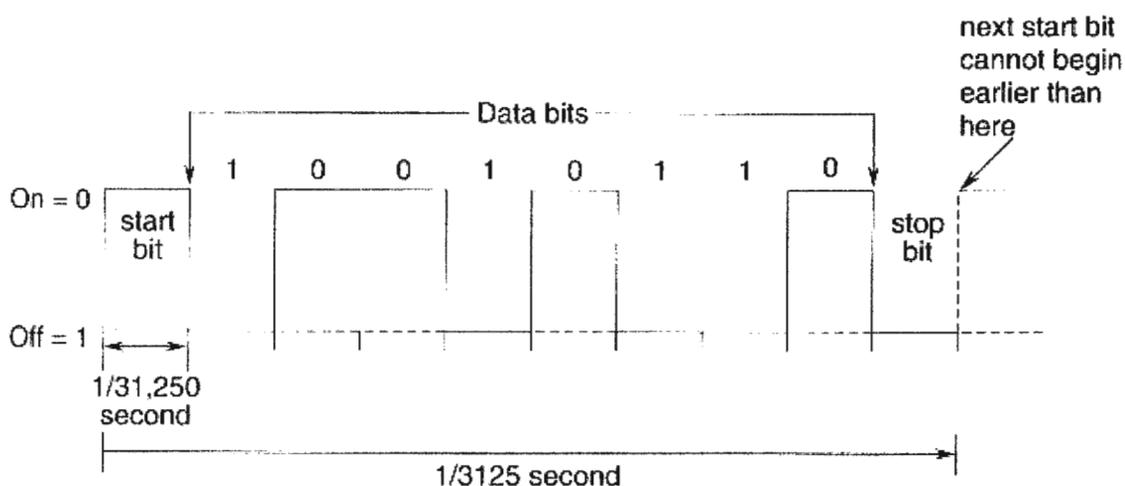
Midi es un protocolo serial de datos, esto significa que envía un bit al mismo tiempo sobre un conductor de cable. Un bit es una simple digito binario, on es 1 y off es 0, estos bits son agrupados dentro de Byte de ocho o diez bits (dependiendo sobre como es usado), estos bytes estan combinados dentro de comandos o mensajes. Un Comando MIDI puede consistir de uno o mas bytes, la especificacion MIDI o tambien llamado "The spec", describe como los comandos son estructurados y decifrados y tambien estipulan las características eléctricas del circuito que crea y interpreta los datos.

Bits y Bytes

La señales MIDI son un pulso, como una onda esparcida, el cual es transmitida 31,250 veces por segundo a través de un pulso de cinco o cero voltios. La especificación MIDI permite un porcentaje de 1% de tolerancia en la velocidad, si es más rápido o más despacio esto no podría trabajar. Esta velocidad es porque 31,250 es un factor de dos poderes (1/32) de 1,000,000. Un millon de ciclos por segundo (1Mhz) o mitad de la taza (500kHz) es comunmente la velocidad del reloj para los circuitos integrados or Unidad de procesamiento central o tradicionalmente llamado CPUs. El CPU es el corazon de una computadora pesonal, de un sintetizador y otros equipos digitales. Dividiendo la taza de reloj y con el factor de dos poderes (1/32) para generar un flujo MIDI es un asunto fácil de lograr a través del software y el circuito diseñado. Las computoras de hoy en día operan en grandes cantidades de multiplos de 1Mhz, lo que idio inicio a escoger una buena idea.

La señal MIDI es asíncrono, lo cual significa que no hay un pulso universal del reloj que abarca todo el tiempo. Un transmisor puede empezar enviar datos MIDI en cualquier momento y el receptor esta esperando entenderlo tan pronto estos son enviados.

Desde que MIDI es basado sobre una señal digital, el pulso tiene dos estados: Lógico 0 y logico 1. (En realidad son "Backwards" electricos: un lógico 0 es un actual encendido pulso y un logico 1 es un actual apagado pulso) El primer bit en cualquier byte MIDI es siempre 0 (on), esto es conocido como el Bit de empieza o "Start Bit". Despues del bit de empieza son ocho bit de datos, los cuales pueden ser unos o ceros, esto son seguidos por un bit de parada o "stop Bit", el cual siempre es uno (off). Por lo tanto, un Midi Byte tiene diez bits, pero solo ocho realmente llevan la informacion. El significado del ultimo bit se refiere que tiene que tener un mínimo porcentaje de tiempo entre un Byte de Midi y el siguiente.



La tasa de Bit de Midi es de 31,125 por segundo y hay diez bits en un Byte, la tasa de Byte de Midi es de 3,125 por segundo. Los mensajes Midi mas completos requieren dos o tres bytes, la tasa de transmision de datos musicales es considerada aproximadamente de 1,000 o 1,500 eventos o mensajes por segundos, esto es muy rapido, pero no es tampoco infinito, tambien esto significa que no puede haber dos mensajes o eventos transmittiéndose al mismo tiempo, en cualquier circunstancia al menos tiene que tener un margen de 0.6 milisegundos, para la mayoria de situaciones esto es sumamente aceptable, pero para pequeña, mínimas o super raras situaciones esto es un factor de limitación del funcionamiento del Sistema MIDI.

Cables y Conectores

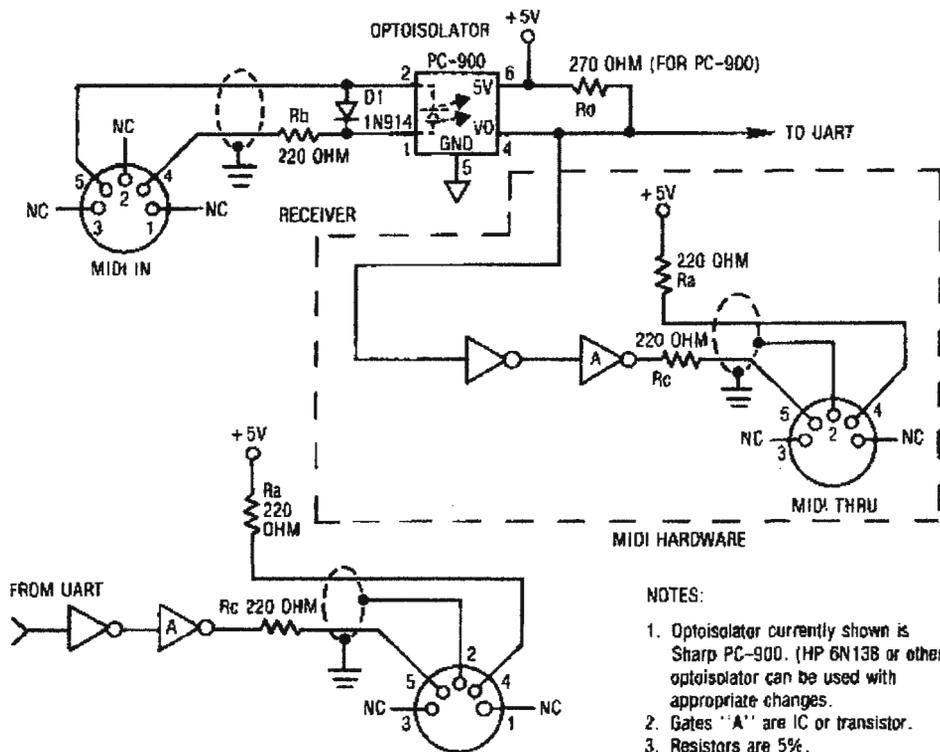
Los dispositivos que transmite los datos MIDI se envían desde un conector MIDI OUT a través de un cable MIDI, que es un par trenzado blindado de dos conductores, igual que el cable de audio, el MIDI OUT es un conector DIN de 5 pines. Los conectores DIN son usados comúnmente en equipo hi-fi europeo.

La señal de MIDI está en el pin 5 del conector, y un voltaje de polarización +5 está sobre el Pin 4. El pin 2 está conectado con el protector de cable, y los pines 1 y 3 no están normalmente conectados.

El cable entonces se conecta en un conector MIDI IN en el dispositivo de recepción, que utiliza los mismos tipos de conexión, excepto el pin 2 del conector que es el protector, no esta conectado. Esto se hace para evitar la posibilidad de lazos de tierra, en los cuales diversos potenciales de tierra de varios dispositivos conectados causan lazo de tierra. En un estudio conteniendo docenas de componentes discretos, lazos de tierra de audio y la energía de los cables son siempre un peligro.

Para prevenir otros posibles problemas eléctricos, el conector MIDI IN no es recomendable atarlo duro con el alambre del dispositivo en el cual se montan. En lugar de esto, todo el conector MIDI IN contiene un aislador óptico, que es un dispositivo electrónico que consiste en una luz minúscula emitida por el diodo (LED) y una fotocélula. Cuando el conector recibe un Bit, el LED se enciende y la fotocélula responde enviando la corriente en el resto de recepción del dispositivo. De esta manera, nunca hay conexión eléctrica directa entre el trazado de circuito de Midi encoding/decoding de los dos diferentes piezas de equipos.

El aislador óptico está conectado con una Chip conocido como uart (universal Receiver/Transmitter Asíncrono). El uart traduce los pulsos de Midi a una forma que el dispositivo de recepción puede entender (o viceversa, en el caso de un transmisor).



Comandos de Midi

Cuando alguien toca un transmisor Midi, o controlador, genera uno o más comandos Midi, o mensajes. Por ejemplo, tocando una nota en un teclado envía un mensaje correspondiente de esa acción al conector Midi out del teclado.

Un mensaje de Midi contiene la información para una acción musical completa. Cuando se presiona una llave en un teclado, genera un mensaje que consista en tres octetos. El primer octeto describe la clase de acción: se ha presionado una llave. El segundo octeto es el número de la llave que se presiona; es decir, cual nota se presiono. El tercero es la velocidad con la cual se ha presionado la llave: la cantidad de tiempo que ha transcurrido entre el comienzo y el final del recorrido de la llave. (Un número de velocidad alta significa que el recorrido tiempo era menos, el cual significa generalmente que la llave fue pulsada más duramente)

El primer octeto se conoce como un byte de comando o byte de estado. (En este ejemplo, el byte de comando es un mensaje NOTE ON) Los segundos y terceros octetos son bytes de datos. La especificación de Midi dicta el significado exacto de todos los bytes de estado. También prescribe el número de bytes de datos que deben seguir cada bytes de estado, sus valores permitidos, y su significado. Algunos comandos de Midi tienen un byte de datos, algunos tienen dos, y algunos no tienen ninguno. Los comandos exclusivos del sistema son seguidos por uno indefinido, y a menudo un gran número de bytes de datos.

La anatomía de un comando de Midi

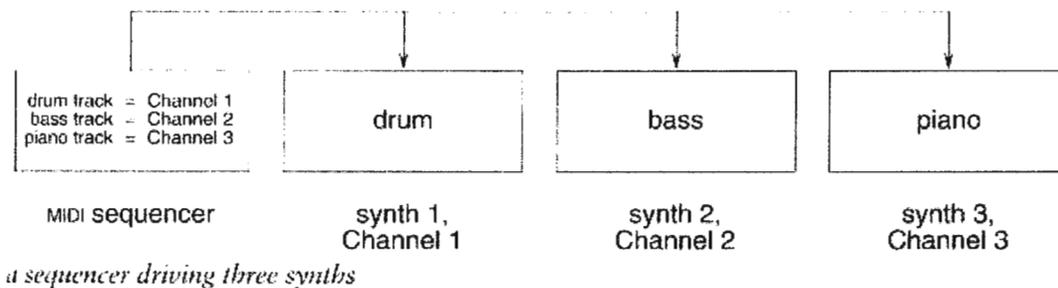
Aquí está un comando típico de Midi en la notación binaria: 10010000 01000101 01100101. En hexadecimal, se observa este comando como: 90H 45H 65H. El primer byte es el comando o el byte de estado. La especificación de Midi indica que cualquier byte de estado que comience con 9 (en hexadecimal) se define como NOTE ON. El segundo octeto es el número de nota. La nota más baja de Midi (OOH) es C, cinco octavas debajo de C media. El número 45H en el ejemplo se traduce en 69 en decimal y la nota sesenta y nueve sobre el fondo C es F sobre el tercer octeto medio de C. El 65H, según la especificación, es la velocidad. En decimal, este número es 101.

Los bytes de estado tienen siempre un valor por lo menos de 80H (128 en decimal). Los bytes de datos tienen siempre un valor entre de 00H y 7FH (0 y 127 en decimal). En los términos binarios, éste significa el bit más significativo o alto bit, en un byte de estado es siempre 1, mientras que el bit alto en un byte de datos es siempre 0. Esto hace la tarea inicial del receptor cuando analiza un octeto recibido de Midi muy simple: Si ve que el bit alto es 1, él sabe que el byte es un byte de estado; si ve que el bit alto es 0, sabe que el byte es un byte de datos y sabe relacionar el byte de datos con el ultimo recibido byte de estado.

Para que un mensaje de Midi sea interpretado correctamente, debe consistir en el correcto número de bytes, según lo dictado por la especificación de Midi. El mensaje puede no ser interrumpido, excepto en algunos casos muy especiales. Si un número escaso de datos de bytes siguen un byte de estado antes de que se reciba el byte de estado siguiente, la recepción el dispositivo asumirá que la información es errónea. En la mejor de las circunstancias, esta clase de error simplemente no se hace caso, pero en algunos dispositivos puede causar extraño comportamiento. Por esta razón, es importante mantener una secuencia de datos limpia de Midi. Un caso especial ocurre cuando el número de los bytes de datos después de un byte de estado son demasiado altos. Esto crea una condición llamada Running status, que discutiremos un poco más adelante.

Canales Midi

La especificación de Midi permite dieciséis canales de los datos. Esto permite múltiples dispositivos Midi reciben cada uno distinta información con solamente una sola línea de Midi conectada. Como el sintonizador en una televisión, cada dispositivo de recepción puede ser fijado para reconocer datos sobre un canal específico de Midi, y no hacer caso a los datos de todos los otros canales.



El Sistema De Comando

La primera mitad (cuatro bits, o primer dígito hexadecimal) del bytes de estado determina naturaleza del comando. Porque el alto bit será siempre un 1, el primer dígito hexadecimal de un byte de estado debe estar entre 8 y F (que representan 1000 a 1111 adentro binario).

Name	Hex Values	Decimal Values	Data Bytes
Note Off	80–8F	128–143	2 (note number, velocity)
Note On	90–9F	144–159	2 (note number, velocity)
Key Pressure	A0–AF	160–175	2 (note number, pressure)
Control Change	B0–BF	176–191	2 (controller number, value)
Program Change	C0–CF	192–207	1 (program number)
Channel Pressure	D0–DF	208–223	1 (pressure)
Pitch Bend	E0–EF	224–239	2 (LSB, MSB)
System Exclusive	F0	240	variable
System Common	F1–F6	241–246	0, 1, or 2
End of System Exclusive	F7	247	0
System Real Time	F8–FF	248–255	0

Notas y programas

9nH es un comando NOTE ON. (n+1 indica el número de canal) Cuando se recibe este mensaje, empieza a tocar la nota. Como vimos anterior, este comando tiene dos bytes de datos: el número de nota y la velocidad. Debido que el bit alto de un byte de datos de Midi es siempre cero, esto nos da una gama de 00000000 a 01111111 en binario, 00H a 7FH, 0 a 127 en decimal, para ambos números de nota y velocidades.

Incluso teclados "primitivos" que no son sensible a la velocidad necesitan enviar un byte de velocidad con su mensajes NOTE ON. El spec de MIDI requiere estos teclados para enviar un byte de la velocidad de 40H.

8wH es un comando NOTE OFF. Cuando se recibe este mensaje, la nota para de tocar o su desarrollo entra la etapa de soltura, si es apropiado. Como NOTE ON, NOTE OFF necesita dos Bytes de datos: el número de nota y la velocidad. no puede parar una nota dada enviándole apenas un comando NOTE OFF, debe enviar una nota Con del mismo número de nota que esta usando en la comando NOTE ON.

Por convención, una NOTE ON seguida por una velocidad de 00 es considerada equivalente a una nota apagada. Algunos dispositivos transmiten solamente la nota On, y la nota On de la velocidad 00 para apagar la nota off. La especificación Midi dice que cualquier método es bueno.

CnH es un comando del cambio del programa (a veces llamado cambio del remiendo). El mensaje del cambio del programa tiene solamente un byte de datos: el número del programa. Un sintetizador digital "programa" es una posición de memoria que contiene todos los valores de los parámetros que determinan el timbre de un instrumento. Un sistema de valores para las formas de onda, frecuencias, encaminamientos de la señal, etc.

Controladores

BnH es un comando de control de cambio (a menudo llamado un controlador continuo, o solo controlador, que no se confundirá con un dispositivo controlador, como una llave tablero). Este comando refiere generalmente a un gesto musical que no sea una nota. Es a menudo algo que puede cambiar el carácter de un sonido después que la nota ha comenzado. Los ejemplos comunes de controladores incluyen la rueda de la modulación, la cual agrega normalmente el vibrato; sostener el pedal (generalmente un interruptor), que mantiene sostener la porción de ir del sobre; y pedal del volumen, que permite nivel cambios o fade-ins y fade-outs. Observar que el volumen de Midi es muy diferente concepto de la velocidad de Midi. El anterior es un parámetro continuo que puede cambiar mientras que la nota continúa sonando,

mientras que el último afecta solamente la nota cuando comienza (o las paradas, en el caso de la nota de la velocidad),

Un mensaje del cambio de control tiene dos bytes de datos: el número del controlador y el valor del controlador en ese momento. Mensajes del cambio de control a menudo ocurren en el flujo

Mensajes De Modo De Canal

Los números controladores 78H (decimal 120) y arriba se llaman Mensajes del modo de canal, porque tienen funciones muy específicas.

El controlador **78H** es el comando todo el sonido apagado. Cualquier dispositivo que reciba esto el mensaje se supone para dar vuelta inmediatamente apagado a todas las notas, y fijó el volumen de todos voces a cero. En un dispositivo de los efectos, este mensaje cortó la cola del retrasa o un reverb, pero no da vuelta apagado al efecto: el nuevo venir de las señales en la entrada audio se procesan normalmente. El segundo byte de datos, siguiendo el número del regulador, es 00. Esto es bastante una nueva adición a especificación Midi, y no se pone en ejecución en muchos dispositivos en fecha todavía.

El controlador **79H** significa reseteo todos los reguladores (en el canal señalado). En otras palabras, fijaron todos los valores del regulador a cero o su ajuste normal (para el ejemplo, volumen se pudo fijar a 7FH). El segundo byte de datos es inaplicable. No cada dispositivo de Midi responde a este mensaje.

El controlador **7AH** es un comando de control local. Esto se determina si el teclado en un sintetizador y el trazado de circuito de sonido generación se liga adentro ¿otras palabras, cuando usted presiona una llave, hace un sonido? cuando control local está encendido (es decir, cuando es el byte de datos después del número del regulador 7FH). No si el control local está apagado (si el byte de datos es 00). Cualquiera otros valores para el último el byte de datos debe ser no hecho

caso (aunque algunos dispositivos interpretan cualesquiera valor diferente a cero como encendido). El propósito de él puede no ser obvio, pero local Controlar es una función muy importante. La capacidad "desempareja" un sintetizador de su teclado (que ocurra cuando el control local está apagado) es crucial en la disposición del multisynth.

El controlador **7BH** significa todas las notas apagado (en el canal señalado). Esto el comando es útil en emergencias cuando las notas se pegan encendido y usted no lo hace saber porqué. Le deja silenciar el sonido rápidamente. Enviando la nota específica apagado los mensajes son siempre la mejor manera de silenciar un sintetizador, pero este comando deja usted logra la misma cosa en una prisa, sin su tener que calcular hacia fuera qué notas están sonando. Una vez más se no hace caso el segundo byte de datos (y no cada dispositivo de Midi responde a este mensaje).

Runing Status.

Normalmente, un mensaje de Midi consiste en un byte de estado y sus byte de datos apropiados. Éstos son seguidos por el byte de estado del mensaje siguiente, y así sucesivamente. Como siempre, la especificación de Midi permite una condición especial en la cual un simple byte de estado se puede seguir por una cadena larga de bytes de datos. Imaginarse que usted esta tocando una llave, y mientras que usted la lleva a cabo, usted varía la presión de crear vibrato. Usted comienza sin la presión, la aumenta al máximo, y entonces lentamente lanzarlo. Esta acción podía producir un total de diversa presión del canal 254 mensajes: 127 para arriba y 127 abajo. Puesto que el byte de estado es siempre igual, y solamente los cambios son de los bytes de datos, si era posible enviar apenas los datos que cambiaban los bytes, reduciría la cantidad de datos por cincuenta por ciento. Runing Status permite que usted haga exactamente esto. Puede ser utilizado con cualquiera de los mensajes descritos hasta ahora, si toman un byte de datos o dos. Runing Status es un "condition" de la secuencia de datos de Midi. Se invoca si se recibe un byte de estado que es seguido

**UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA EN COMPUTACION**

RECTOR:

ING. FEDERICO MIGUEL HUGUET RIVERA

SECRETARIO GENERAL:

LIC. MARIO OLMOS

DECANO DE LA FACULTAD DE INGENIERIA:

ING. ERNESTO GODOFREDO GIRON

ASESOR DEL TRABAJO DE GRADUACION:

ING. CARLOS PUENTE

TUTOR DEL TRABAJO DE GRADUACION:

ING. JAIME ANAYA

JURADO EVALUADOR:

ING. JORGE LOPEZ

LIC. OSCAR MENDEZ

ING. ARNOLDO RIVAS

El Runing Status se puede también utilizar en el envío de una cadena larga de notas pero solamente si envían el comando Apagado de la nota como nota Qns con la velocidad cero. Aquí está un ejemplo de un cadena de dos notas que podrian ser enviados sin usar Runing Status.

90 45 31 80 45 40 90 62 7A 80 62 40

Esta serie de comandos contiene la información siguiente: Observa la nota on 45H con la velocidad 31H, la misma nota off con la velocidad 40H, observa note on 62H con velocidad 7AH, la misma nota off con la velocidad 40H. Gracias al Runing Status, la secuencia se puede ajustar al siguiente.

90 45 31 45 00 62 7A 62 00.

Mensajes Del Sistema

Los comandos discutidos hasta ahora se conocen colectivamente como mensajes de canal de voz, porque contienen un número de canal y su voz al mismo tiempo.

Los comandos permanentes de la especificación Midi se llaman mensajes del sistema. Un mensaje de sistema no contiene un número de canal, y su primer dígito será siempre F.

Exclusivo del Sistema

Un mensaje que comienza con FOH se llama un exclusivo del sistema o un Mensaje SysEx. Este tipo de mensaje es hecho específicamente para modelos de dispositivo de MIDI que reciben la información que otros dispositivos en la línea no harán caso (incluso si están en el mismo canal de MIDI).

Aquí está un uso simple de los comandos exclusivos del sistema: Un conjunto de Parámetros que describen un programa, dicen un sonido del piano, en cierto modelo de sintetizador. Usted tiene otro sintetizador que sea el mismo modelo, y usted desea

ese sonido del piano este disponible en el segundo sintetizador también. Usted podría mirar todos los parámetros en el primer sintetizador, anotarlos, y entonces entrarlos uno a la vez en el segundo sintetizador. O, usted puede mandar el primer sintetizador arreglar todos los parámetros en un mensaje exclusivo del sistema y enviar el mensaje por el cable de Midi al segundo sintetizador. Si el segundo sintetizador esta configurado aceptar los mensajes exclusivos del sistema, él recibirá y almacenar los parámetros en uno de sus registros del programa. Cuando usted llama en ése registro en el segundo sintetizador, él tocara el mismo sonido del piano. Porque los mensajes describen un sonido para cierto tipo de sintetizador, otros modelos en la línea no harán caso del mensaje totalmente. Este tipo de intercambio del parámetro de lista es llamado un sistema descarga exclusivo o descarga del parámetro. Una descarga puede contener parámetros de una sola voz o los parámetros de todas las voces de memoria del sintetizador. Así un mensaje exclusivo del sistema puede contener centenares o aún millares de bytes. Un mensaje exclusivo del sistema tiene un número de identificación especial llamado identificación del fabricante en el byte inmediatamente después del encabezado FOH. Cada fabricante de equipo Midi es publicado un número único de identificación que es administrado por la asociación de fabricantes Midi (MMA) y el Comité De Estándares de Japón Midi (JMSC). Por ejemplo, el número de la identificación de Yamaha es 43H, y por lo tanto, cualquier mensaje exclusivo del sistema diseñado para el equipo de Yamaha comienza con la identificación de FO 43H. Casio es 44H, el léxico es 06H, etcétera.

Con el crecimiento explosivo de Midi después de su inicio, era evidente que tan pronto que aparecieron estos números de solo un byte alborotarian a todo fabricante y por eso un nuevo sistema de identificaciones de fabricante fue publicado con tres octetos, el primer siempre esta 00H. Las Identificaciones se dividen geográficamente como se explica a continuación (todos los valores son hexadecimal).

Las compañías americanas consiguen a números 00 00 01 a 00 5F 7F, las compañías europeas y australianas consiguen 00 20 00 a 00 3F 7F, las compañías japonesas consiguen 00 40 00 a 00 5F 7F y las compañías en otras áreas consiguen 00 60 00 a 00 7F 7F.

Qué viene después de la identificación del fabricante es enteramente del fabricante. La única restricción es que ningunos de los bytes tengan valores más arriba que 7F, con el objetivo que el dispositivo que reciba nunca se confunda con un nuevo byte de estado. No hay running status cuando reciben los mensajes SysEx y no puede ser interrumpido por ningún otro mensaje. Si se envía un byte de estado de alto valor durante un mensaje de SysEx, se termina el mensaje.

Muchos fabricantes utilizan sub-Identificación para identificar diversos modelos, y/o las identificaciones del dispositivo para tratar los dispositivos de diversos canales Midi. Por ejemplo, usted puede ser que tenga un sistema con ocho sintetizadores idénticos, pero si usted desea enviar a nuevo sonido a solamente uno de ellos, usted puede distinguirlo por su dispositivo de la identificación. Las identificaciones no necesariamente tienen que corresponder con los canales de Midi; puesto que usted puede utilizar un byte completo para una identificación del dispositivo, usted puede tratar 128 diversos dispositivos, no solo dieciséis. Esto es particularmente provechoso cuando usted está trabajando con sintetizadores multilimbricos (que reciben en los múltiples canales de Midi) o cuando usted tiene usando un sistema muy grande de cables múltiples de Midi.

Para finalizar cualquier mensaje del sistema exclusivo es llamando un byte F7H. Este byte es abrevia también llamada End del sistema exclusivo, y a veces en algunas documentaciones se llama como EOX. Cuando se recibe un F7H, todos los dispositivos en el sistema que no hacían caso del mensaje del sistema exclusivo despiertan y prestan atención a lo que viene después. Hasta que se recibe el F7H, un mensaje del sistema exclusivo no se puede interrumpir.

Los mensajes del sistema exclusivos tienen una amplia variedad de uso. Son comúnmente usados para que los sintetizadores sean programados a través de la

computadora, usando el editor de programas. Esta capacidad ha causado la introducción de los sintetizadores con muy pocos controles del panel de delante, o aún ninguno. Éstos los sintetizadores se pueden programar solamente de una fuente externa de Midi, tal como computadora u otro sintetizador.

El fabricante que cuenta con una identificación de mensaje del sistema exclusivo tiene la exclusiva de decidir cómo sus mensajes van a ser enviados y a ser interpretados, pero un dato importante es que no los posee. Cualquiera persona puede utilizar los mensajes sin el permiso del fabricante, de tal manera no está en conflicto con la manera que el fabricante los utiliza. De hecho, se requiere que el fabricante publique el formato completo de SysEx para cada nuevo dispositivo que introduce (aunque algunos fabricantes son poco lentos en este aspecto). Esto significa que cualquier persona puede escribir un programa de computadora, o inventar un nuevo tipo de controlador, que trata directamente los parámetros de SysEx del cualquier sintetizador Midi o de otro dispositivo.

Tiempo real Del Sistema

Los mensajes en tiempo real del sistema se utilizan para sincronizar el equipo mientras este funcionando. Los mensajes en tiempo real del sistema no tienen ningún número de canal y no requieren bytes de datos. Pueden ser insertados en la secuencia de datos de Midi en cualquier momento, incluso en la mitad de otro mensaje. Según lo mencionado anterior, pueden incluso pueden enviarse dentro del Running status sin interrupciones (es decir, no están interpretado como nuevos bytes de estado).

Aquí están los valores definidos del tiempo real del sistema. (F9H y FDH son indefinidos actualmente.)

Mensajes Del Tiempo real Del Sistema

Nombre	Valor Hexadecimal	Valor Decimal
Tiempo de reloj	F8	248
Start	FA	250
Continuar	FB	251
Stop	FC	252
Detección Activa	FE	254
Reinicio Del Sistema	FF	255

F8H es un mensaje de tiempo. Los relojes que miden el tiempo se utilizan para que dos o más dispositivos de Midi trabajen al mismo tiempo. Estos mensajes son generados veinticuatro veces por la cuarta nota de modo que su frecuencia varíe con el tiempo. Un dispositivo Maestro envió los relojes que miden el tiempo y los dispositivos esclavo le siguen.

El reloj que mide el tiempo también se llama MIDI Sync (aunque no aparece este término dentro de la especificación de Midi). El término Midi Sync es quizás más apropiada cuando es utilizado en un sentido más amplio, abarcando los comandos del Start, Stop, y continuar. De esta manera, cuando se habla de un dispositivo que es compatible Midi Sync, significa que el dispositivo responde a todos estos comandos.

FAH es un mensaje de comienzo (Start). Esto le dice a un dispositivo ir al principio de su canción y comenzar a tocar en el tiempo determinado por los mensajes tiempo que siguen inmediatamente.

FBH es un mensaje de continuación, que es similar comenzar (Start), excepto que el dispositivo que recibe tocara su localización actual, no necesariamente desde el principio.

FCH es un mensaje de parada (stop). Esto le dice a un dispositivo parar de tocar y esperar a Comenzar o continuar (no apenas un reloj que mide el tiempo).

FEH es un mensaje de detección activo. La detección activa es una característica opcional de transmisores de Midi. Cuando se utiliza esta característica, un mensaje se envía cada 300 milisegundos o menos. Este comando fue diseñado para prevenir las situaciones en las cuales un sintetizador recibe una nota sobre mensaje y entonces pierde de alguna manera su entrada antes de que la nota reciba su correspondiente apagado. Esto situación da lugar a una nota que nunca pare (conocido como "nota pegada"). Si el dispositivo nunca recibe un mensaje de detección activo, él se comportará como si nunca a existido. Sin embargo, una vez que un dispositivo reciba su primer FE, cuenta con uno cada 300msec, si falta uno, para todo el sonido. La detección activa raramente se utiliza, porque agrega una cantidad pequeña de gastos indirectos adicionales a los datos de Midi. Los datos de detección activos pueden también confundir algunos dispositivos cuando son la recepción de bloques grandes de datos.

FFH es un mensaje del reinicio del sistema. Esto manda a todos los dispositivos para volver a su condición de ciclo inicial; es decir, apagar todas las notas, parar el enviar de los relojes, reajustar todos los controladores, etcétera. Este mensaje también raramente se utiliza.

Modos

En los primeros días de MIDI, los modos eran un tema muy grande, y una descripción de ellos fueron encontrados al principio de cada discusión de la especificación de Midi. La llegada de sintetizadores polifónicos, multitimbricos y samplers ha hecho los modos menos importantes de lo que era antes. Sin embargo, Hay situaciones dentro del cuál pueden ser útiles. Hay cuatro modos Midi. Éstos se configuran generalmente en un panel delantero de los sintetizadores, o usando dos interruptores continuos del controlador: Omni on/off (Controladores 7DH y 7CH) y Poly On/Mono on (Controladores 7FH y 7EH). Los bytes de datos de todos estos comandos (excepto a veces mono on) se suponen a ser irrelevantes, pero algunos sintetizadores requieren valores diferentes a cero (o aún 7FH) para actuar con ellos.

En modo Omni on, un dispositivo de recepción responderá a los datos de todos los canales Midi. En el modo Omni, responderá en apenas un canal o en el caso de un sintetizadores multitimbrico, múltiples, pero específico canales. En el modo Poly, un dispositivo responderá a las múltiples entrantes notas polifónicas. En modo Mono, responderá monophonically; es decir, solamente sonará una nota a la vez no importa cuantas intentan tocar.

Hay cuatro modos que corresponden a los cuatro estados posibles de interruptores del controlador.

- Omni On/Poly: también se conoce como modo 1, útil en situaciones con un transmisor MIDI y un número pequeño de receptores, todas las notas pueden ser tocadas simultáneamente sin distinción por número de canal.
- Omni On/Mono: o el modo 2 muy raramente se utiliza.
- Omni Off/Poly: o el modo 3, es en gran medida el modo más común.
- Omni Off/Mono: o el modo 4.

Entonces, ¿Qué es MIDI?

Con el crecimiento del uso de MIDI surgieron muchas confusiones, entonces que es MIDI y que puede hacer. Aquí están algunas listas que ayudan a clarificar que es y que no es MIDI.

MIDI es

- Una manera de interconectar los instrumentos musicales, las computadoras, y otros dispositivos
- Una manera de tocar sonidos sintetizados en una tarjeta de sonidos
- Un protocolo de comunicaciones estandar
- Una manera de controlar los dispositivos no musicales, tales como iluminación del teatro, registradores de cinta, etc.

MIDI es

- Concerniente al funcionamiento de la música pero no al audio
- Una representación compacta de un funcionamiento musical
- Cambiable: un funcionamiento ordenado se puede editar fácilmente para cambiar las notas, el tiempo, e incluso los sonidos hechos por un instrumento

MIDI no es

- Audio: solamente un instrumento de recepción de MIDI que hace sonido
- Audio de la forma de onda: grabadas en las tarjetas de los sonidos y para tocar audio de forma de onda
- Una panacea: algunas tareas son hechas usando audio, mientras que otras tareas es mejor utilizar MIDI

MIDI no es

- Solamente para las tarjetas de los sonidos de las computadoras
- Retrasos para la buena sincronización: Los acontecimientos de MIDI se envían muy rápidamente, bastante aceptable para la mayoría de las secuencias al tocar con la sensación humana apropiada
- Infinitamente rápido: también muchas notas pueden estorbar los trabajos y causa retrasos

Una fuente de confusión sobre las posibilidades de Midi es la combinación del audio de la forma de onda, síntesis y tecnología MIDI. Muchas tarjetas de sonidos y los sintetizadores crean sonidos realistas usando síntesis de tabla de onda. En un dispositivo de sonidos, una tabla de onda es un permanente bloque de datos de audio digitales, almacenado a menudo en una ROM. Este audio está subdividida en pequeño porciones.

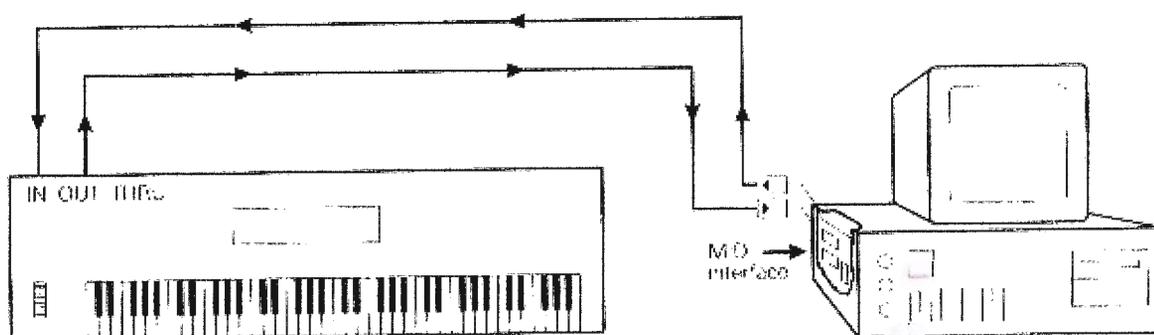
Cuando el dispositivo recibe las notas de MIDI, toca las muestras audio digitales que corresponden a esas notas particulares.

Hay un sin fin de combinaciones de maneras de interconectar los dispositivos de MIDI, limitadas solamente por el presupuesto del usuario para los instrumentos MIDI. La manera más básica de utilizar MIDI es conectar dos instrumentos usando dos cables de MIDI. La mayoría de los instrumentos tienen un puerto de salida de MIDI OUT y un puerto de entrada MIDI IN.

Los datos de MIDI son enviados por una salida de los dispositivos y a través los puertos, y son recibidos por su puerto de entrada. El puerto directo, que no aparece en todos los instrumentos, hace salir simplemente exactamente qué es recibida por el puerto de la entrada de los dispositivos. Los cables de MIDI conectan siempre un puerto de salida de MIDI con un puerto de la entrada de MIDI. Evitar conectar entrada a la entrada o a la salida a la salida. Conectando un MIDI entrado con otro MIDI la entrada no lastimará cualquier cosa, pero el sistema no trabajará correctamente.

Conexiones Simples

Conectar los teclados y los módulos de los sonidos es fácil y práctico, pero es más interesante conectar los dispositivos de MIDI a una computadora y utilizar la computadora para procesar los acontecimientos de MIDI. Siguiente figura demuestra la manera más simple de conectar un teclado con una PC.



Aquí, la computadora tiene un interfaz MIDI que está en una tarjeta de sonido dentro de unas ranuras de extensión de la computadora. Este interfaz puede ser parte de una tarjeta de sonido o puede ser una tarjeta de hardware integrada. Este ejemplo demuestra dos conexiones, una entrada y una salida. Pero hay otras opciones. Por ejemplo, hay muchas diversas clases de los interfaces de solo un puerto (uno de entrar y uno de salir) y el multipuerto interconecta dos o más entradas o salidas disponibles. Algunas de ellas son tarjetas internas, como la que se muestra en el gráfico. Otros son las cajas externas que conectan con los puertos de impresora de la PC o los puertos seriales. Hay interfaces uniforme clasificados que tienen ocho entradas separadas y ocho salidas separadas. Pero las interconexiones de MIDI son básicamente iguales si la computadora tiene uno u ocho puertos.

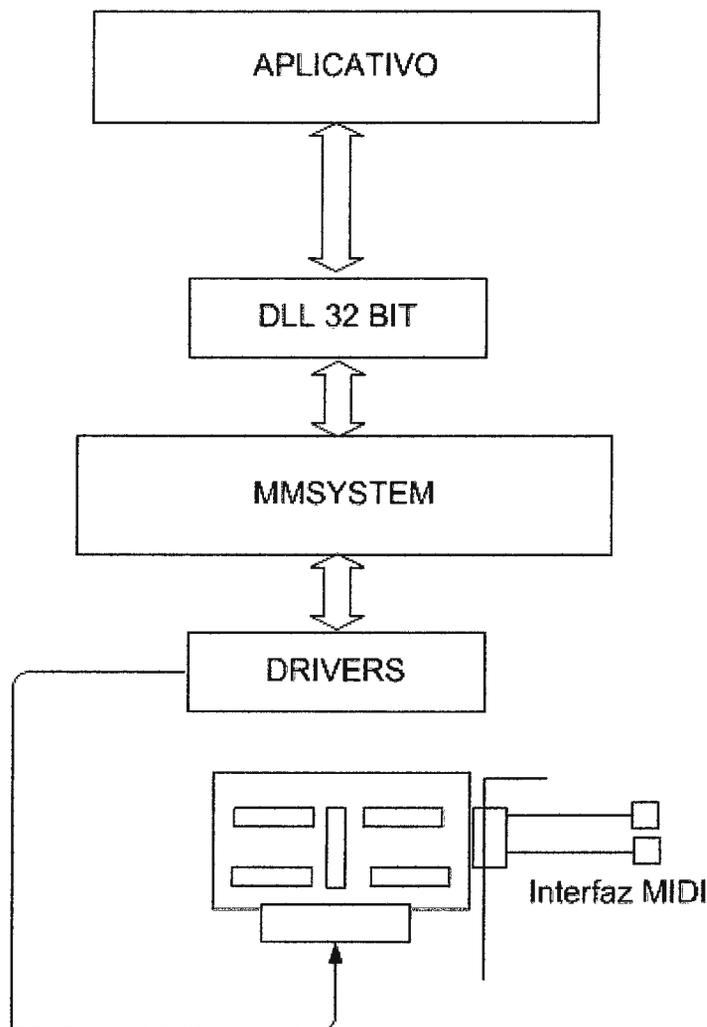
Muchos programas de MIDI utilizan las funciones MIDI disponibles en tarjetas de sonidos. En este caso, no hay conexión externa de MIDI. La computadora encamina datos de MIDI directamente a la tarjeta interna donde el sonido es generado por el sintetizador. Una computadora dada puede tener cualquier combinación de MIDI interno y externo dispositivos.

Las conexiones demostradas aquí rasguñan solamente la superficie. Hay una amplia variedad de dispositivos MIDI disponibles, cada dispositivo que presenta oportunidades y compensaciones únicas. El asunto de cómo utilizar todos los equipos MIDI está más allá del alcance de esta tesis.

PROGRAMACION EN WINDOWS

Windows tiene dos diferentes pero relacionados maneras de manejar MIDI. Uno es interfaz de programación de alto nivel, llamado interfaz de controlador de medios (MCI), que incluye la capacidad llamada playback del secuenciador MCI. El secuenciador MCI proporciona simple playback de una secuencia MIDI, no es capaz de guardar o sincronizar otros procesos o fuentes de sincronización. Sin embargo, su simple uso es ideal para algunos programadores, ya que es todo lo que necesitan para utilizarlo. Afortunadamente, este componente de Windows esta completamente documentado en los manuales que acompañan la mayoría de las herramientas y de los compiladores específicos de desarrollo de Windows.

El otro método para manejar los datos de MIDI se incluye en un sistema de funciones de bajo nivel, esto abarca la base del subsistema de los multimedia de Windows. Cualquier función imaginable de MIDI se puede escribir usando estas funciones de uso general. De hecho, el secuenciador del MCI fue creado usando estas llamadas de función. Pero usar tales funciones de bajo nivel puede ser muy difícil y puede llegar hasta la frustración.



Afortunadamente, todos los detalles "repugnantes" han sido tomados con cuidado con la herramienta de Midi. Las aplicaciones solamente necesitan hacer una simple llamadas a la herramientas MIDI para poner las características de MIDI en ejecución. Windows incluye otra característica de programación MIDI. El "streams" provee otro método de tocar los acontecimientos de MIDI. Para utilizar el "streams", las aplicaciones colocan los eventos MIDI en un Buffer, incluyendo la información de la sincronización para cada acontecimiento. Las funciones un sistema simple aceptarán los datos y tocará la secuencia. Estas funciones son similares a funciones de bajo nivel proporcionadas por la caja de herramientas de MIDI. Sin embargo, como el secuenciador MCI, el "streams" Solamente puede tocar la secuencia, limitando su utilidad para los usos completamente equipados de MIDI.

Las funciones de la caja de herramientas de MIDI proporcionan las características fáciles de utilizar como grabar y de tocar múltiples secuencias MIDI, archivar en disco con el estándar de MIDI, sincronización de reloj de MIDI y mucho más.

El MMSYSTEM nació desde las primeras versión de Microsoft Windows, en la que se implemento en un módulo de código de 16-bit y todos los drivers del dispositivo MIDI eran también de código de 16-bit, en Windows 95 algunas porciones de drivers eran de 32 bit, estas se llamaban VXD, y para llamar a las funciones de 16 bit del MMSYSTEM, usaban un proceso llamado Trunking, esto permite una comunicación viceversa entre el código de 32 y el de 16 bits, ahora en las versiones actuales de Windows la mayoría de códigos son de 32 bit.

El código del aplicativo de captura musical es de alto rendimiento de MIDI, cuya finalidad es el de composición, experimentación musical y educación. El código esta hecho en clases de C++ que utilizan las funciones proporcionadas por el DLLs de Windows. Estas clases implementan la entrada de MIDI, la salida de MIDI, la sincronización, SMF, la pista, el menú de dispositivo, el editor de eventos MIDI y el editor de Solfa, toda estas clases están basada en la biblioteca de clases MFC(Microsoft Foundation Class).

Para el uso de estas clases no es necesario entender de qué está compuestos las DLLs Windows. Este DLLs se puede llamar de cualquier programa de C o de C++, y se comunican a través de mensajes de Windows, mas adelante se explicara como interactuar con la DLL de Windows

Windows y DLLs

¿Que son DLLs y porqué lo necesitamos?

Una librería de enlace dinámico o DLL, es un módulo ejecutable que contiene funciones que se pueden llamar desde cualquier programa. Toda estas llamadas lo unico que necesitan saber es el nombre de la DLL y qué funciones están disponibles (exportaciones) para su uso. Estas DLL encapsulan sus definiciones y dan a conocer solo las necesarias para su uso. Más de una aplicación puede llamar funciones de una DLL. En Windows hay normalmente una copia de código DLL que es compartido por todos los aplicativos.

Un programa no necesita saber todo sobre las funciones en el DLL, con excepción de nombres y los parámetros de las funciones que llama y por supuesto qué hace con tales funciones. Mientras que el programa está funcionando, el DLL no se puede incluso cargar de disco hasta que sea necesario. Esto ahorra memoria y ayuda a ejecutar los programas más rápidamente. No necesita saber los detalles profundos como la dirección de memoria donde reside una función dada.

Mientras que un dispositivo MIDI está abierto, Windows envía los mensajes que indican las cosas interesantes están sucediendo. Por ejemplo, cada vez que un mensaje MIDI es recibido por un interfaz MIDI de hardware, el mensaje es enviado a un driver del dispositivo. El driver de dispositivo entonces envía Datos de MIDI a cualquier aplicación que haya abierto el dispositivo.

Los mensaje MIDI podrán ser recibido en cualquier momento, aunque Windows está en medio de algo crítico, como tener acceso a un dispositivo de almacenamiento o asignación de memoria. El driver del dispositivo llamará al aplicativo (que es, cualquier que abrió el dispositivo) en el tiempo de interrupción. Si esta parte del código del aplicativo se pagina hacia fuera del disco, el sistema no poder recargar el código de disco y lo más probable es que exista una falla.

Consecuentemente, el código que se llamará por el driver del dispositivo debe residir en una parte fija de memoria, en un segmento de paginación bloqueada. Cualquier dato global que el código espera debe de tener acceso debe de ser fijo y bloqueado también. La manera más fácil y la mejor manera de fijar el código es colocarlo en una DLL y configurar los atributos FIJO en el archivo de la definición de la DLLs. De esta manera, el DLL será cargado por el sistema y estará bloqueado de modo que nunca será trasladado o paginado fuera de memoria. En cualquier momento el driver del dispositivo necesita llamar el código de la DLL, él sabe que código presente necesitará.

CAPITULO VI

DESARROLLO DEL DISEÑO

1. DISEÑO DEL SISTEMA

1.1. DISEÑO DE MENUS

Menú Archivo: Opciones para trabajar con un archivo de almacenamiento MIDI

Nuevo : Abre un nuevo proyecto colocando una nueva plantilla de diseño

Abrir: Carga un diseño y configuraciones guardadas en disco

Guardar: Guarda en disco el la captura de notas musicales en formato MIDI.

Guardar Como...: Guarda con otro nombre un archivo de formato MIDI que estaba ya guardado en disco

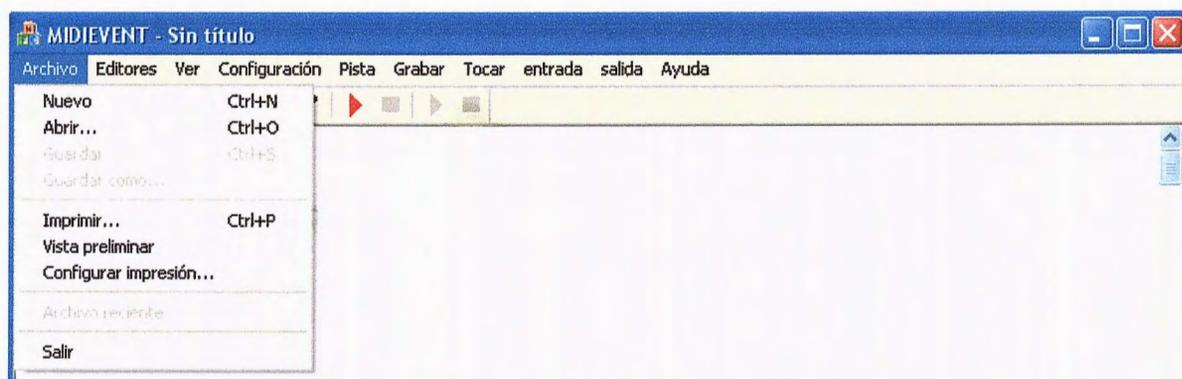
Imprimir: Abre la ventana de configuración de impresión.

Vista Preliminar: Abre la ventana de configuración preliminar de la impresión.
estaba ya guardado en disco Configuración

Impresión: Muestra y tiene Acceso a los archivos mas recientes

Recientes: Muestra y tiene Acceso a los archivos mas recientes

Salir: Salir completamente del Aplicativo



Menú Editores: Editores para los eventos MIDI y notas musicales.

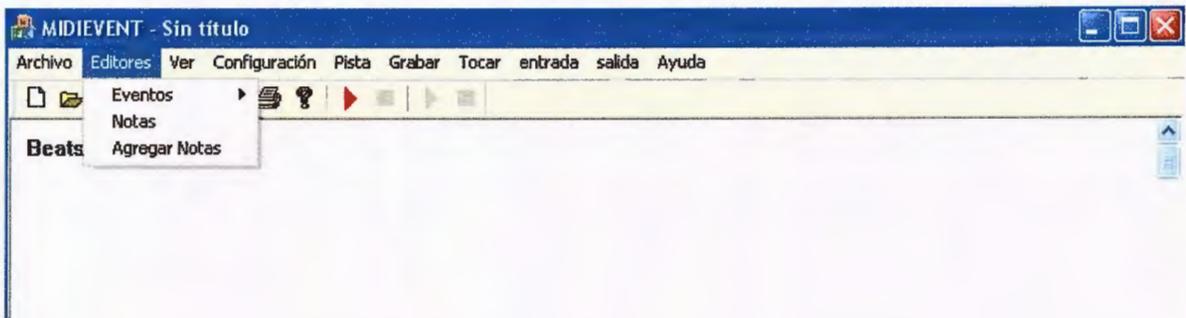
Eventos: Muestra los eventos MIDI en forma decimal y Hexadecimal

Normal: Muestra los eventos Midi en forma Decimal

Hexadecimal: Muestra los eventos Midi en forma Hexadecimal

Notas: Muestra las notas Musicales en forma grafica

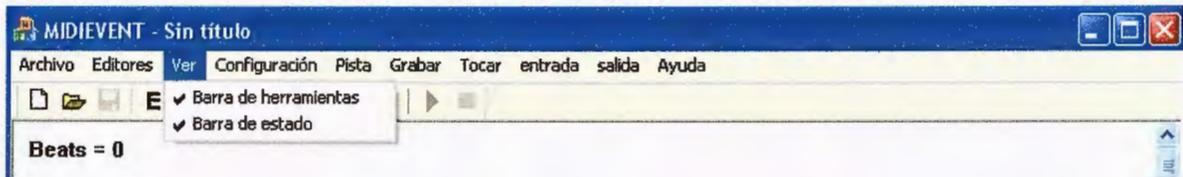
Agregar Notas: Agrega eventos midi.



Menú ver: muestra y esconde la barra de herramienta y barra de estado

barra de herramienta: Muestra y esconde la barra de herramienta

barra de estado: Muestra y esconde la barra de barra de estado



Menú Configuración: Opciones de configuración de la captura notas musicales

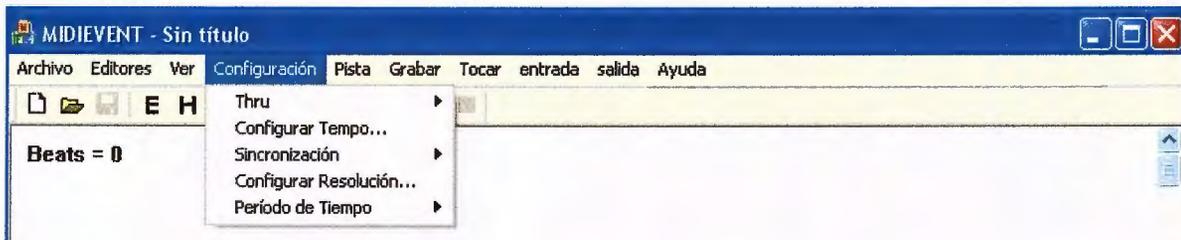
Thru: Habilitar y deshabilitar el Thru, si la interfaz MIDI cuenta con este puerto

Configurar Tempo: Cuadro de texto que especifica el los Beats por minutos

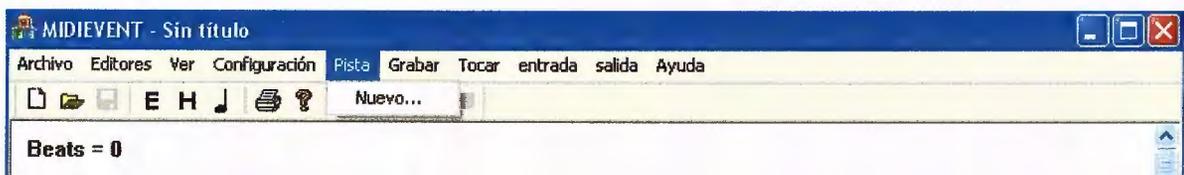
Modo de Sincronización: Cuenta con dos opciones de sincronización interno y de MIDI

Configurar Resolución: Cuadro de texto que especifica el numero de resolución por Beats

Periodo de tiempo: Cuenta con dos opciones de tiempo de 1 Milisegundos y 10 Milisegundos

**Menú Pista:** Opciones para con los datos almacenados en la plantilla de pista o track.

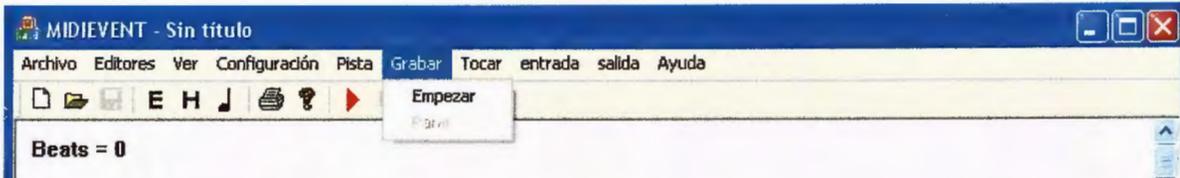
Nuevo: Crea nuevo repositorio de pista o track



Menú Grabar: Opciones para capturar los mensajes midi.

Empezar: Empieza a capturar los mensajes midi

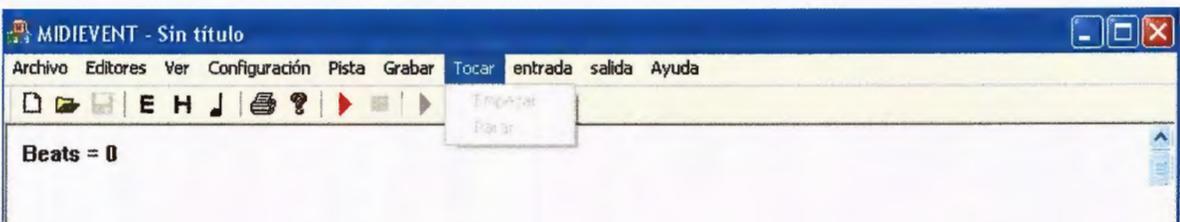
Parar: Para de capturar los mensajes midi



Menú Tocar: Opciones para tocar los mensajes midi.

Empezar: Empieza a tocar los mensajes midi

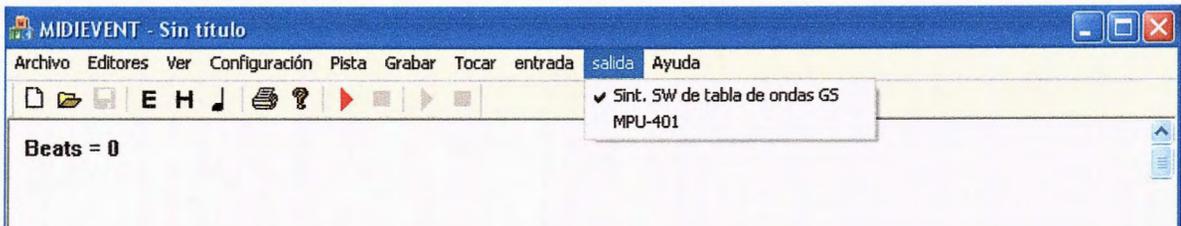
Parar: Para de tocar los mensajes midi



Menú Entrada: Muestra los dispositivos de entrada conectados a la interfaz MIDI y el usuario tiene que seleccionar uno.



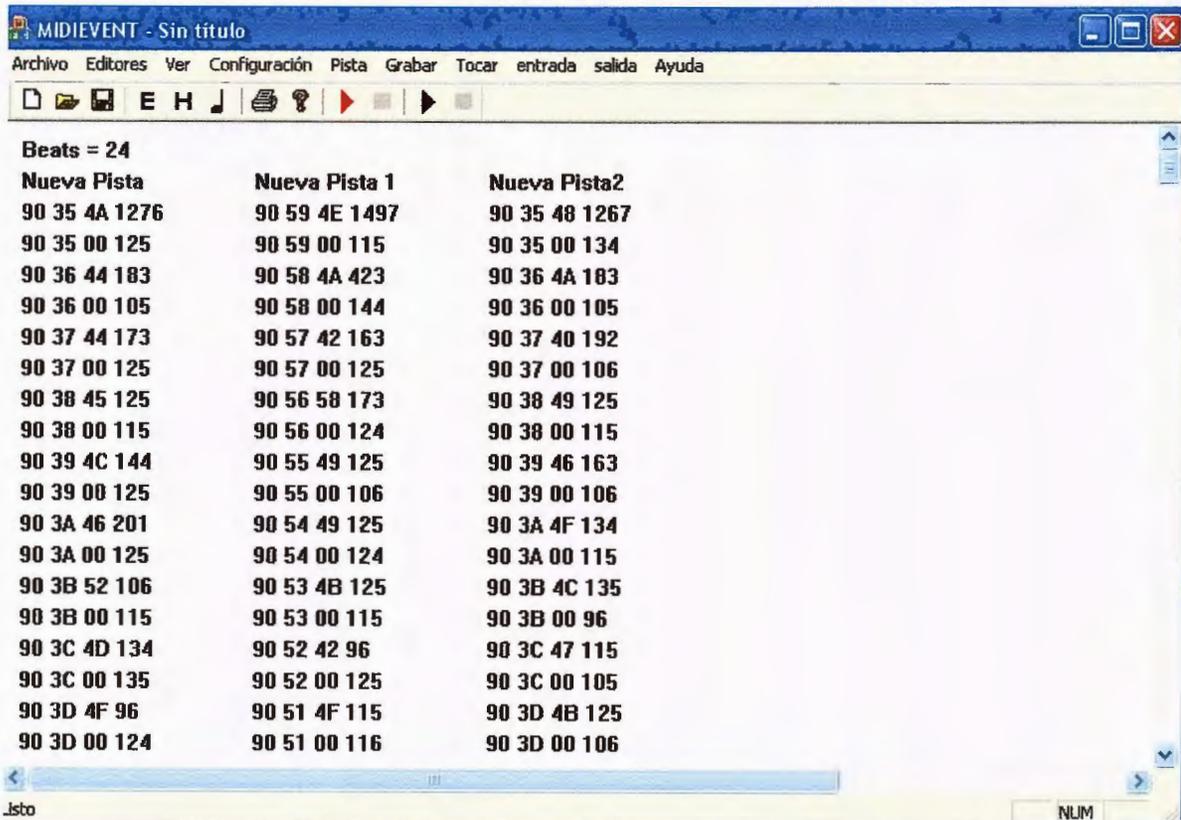
Menú Salida: Muestra los dispositivos de salida interna (Tarjeta de Sonido) e externa de los equipos MIDI.



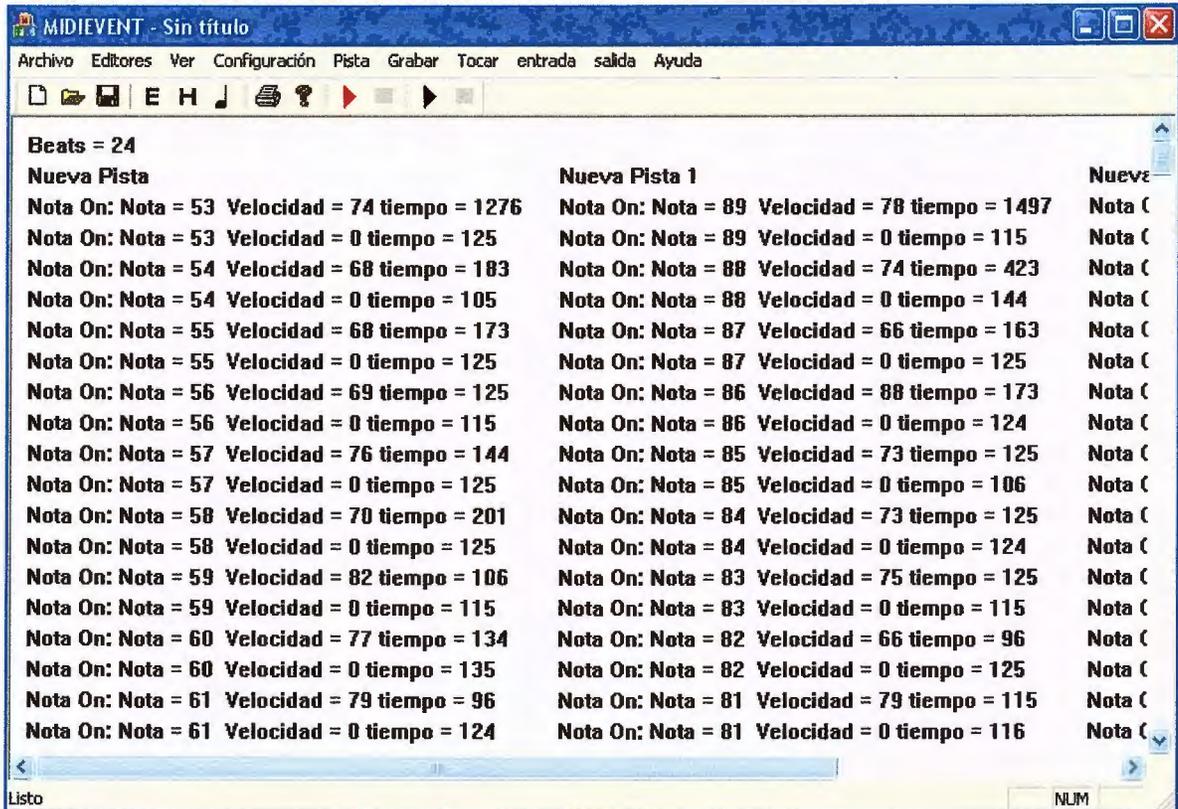
1.2. DISEÑO DE PANTALLAS

Hay tres tipos de plantillas los cuales muestran los eventos midi capturados en formato decimal, hexadecimal y en formato gráfico mostrando las notas musicales.

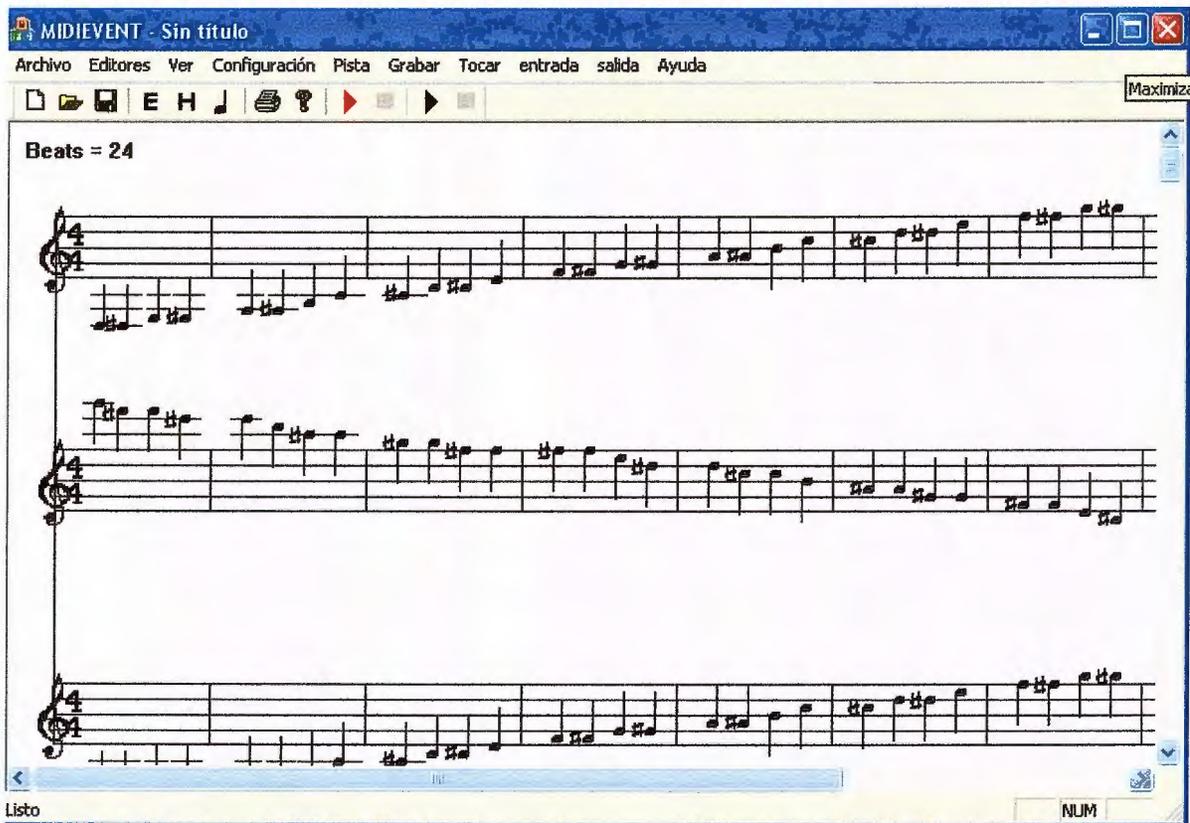
1.2.1. Pantalla de captura de eventos en formato hexadecimal: mostrará los eventos midi capturado o almacenados en formato hexadecimal



1.2.2. Pantalla de captura de eventos en formato decimal: mostrará los eventos midi capturado o almacenados en formato decimal



1.2.3. Pantalla de captura de notas musicales: mostrará los notas musicales en forma grafica capturados por los eventos MIDI.



1.3. ESTRUCTURA DE ARCHIVOS

El archivo de almacenamiento es en formato SMF de MIDI. La especificación define un formato de archivo que contiene el tiempo estampado de dato Midi, junto con eventos tocables en forma de meta eventos. Los archivos del estandar MIDI son diseñados para plataforma independiente. Los archivos creados en este proyecto pueden ser transportados en otro sistema y en otro programa y ser leídos y tocados sin ningun problema, asi como también el aplicativo es capaz de leer, tocar y editar cualquier archivo MIDI.

El Aplicativo de captura de notas musicales contiene rutinas que permiten leer y escribir archivos estandar MIDI, Los eventos son almacenados en los archivo Midi a través de pistas o tracks.

Un archivo de estandar MIDI esta organizado dentro de una serie de bloques llamados "Chunks". Todos los "Chunks" tienen la misma estructura que son:

- Una cadena de cuatro caracteres identificando el type de un particular "Chunk"
- Un valor de 32bit que especifica el numero de bytes de datos en el "Chunk".
- Los siguientes bytes de datos del "chunk".

Los "chunk" no estan jerarquizados, son planos y secuenciales, van uno tras uno.

CONCLUSIONES

El proyecto planteado facilitará el aprendizaje de un usuario en lo que respecta a la tecnología MIDI y Música en General.

- El aplicativo es capaz de capturar los eventos Midi.
- El aplicativo es capaz de mostrar los eventos MIDI capturados o almacenados en los archivos MIDI en forma de lista de eventos decimal, hexadecimal y en forma grafica las notas musicales de los eventos MIDI.
- El aplicativo es capaz de tocar, almacenar y editar archivos SMF Midi.
- El aplicativo es capaz de agregar pistas o track para luego tocarla, almacenarla y editarla, independientemente de otras canciones MIDI ya capturadas o guardadas.
- El aplicativo es capaz de imprimir Tanto las notas musicales en pantalla como los eventos MIDI en formato decimal y hexadecimal.
- El aplicativo es capaz de Configurar el tiempo de la cancion MIDI atravez del tempo (Beat por minuto), resolucion (Tick por Beat) y interrupción del reloj (1 y 10 Microseguntos).
- El aplicativo es capaz reproducir las notas musical capturadas o almacenadas a traves de la tarjeta de sonido de la computadora o a través de un instrumento MIDI.
- El proyecto tiene un gran uso en el área de educación Musical.
- El Proyecto permite Componer una melodía Musical a través de un equipo Musical con el estándar MIDI.
- El Proyecto contiene suficiente documentación para poder ser mejorado, según lo requieran las exigencias en el futuro, tales como notación solfa.

BIBLIOGRAFIA

Libros de Texto

- Análisis y Diseño de Sistemas de Información, James A. Senn, Segunda Edición.
- Basic MIDI, Paul White.
- Midi for the Professional, Paul D. Lehrman and Tim Tully.
- Manual de Referencia Visual C++, Chris H. Pappas y William H. Murriay.
- Programación Avanzada con Microsoft Visual C++, David J. Kruglinski.
- Microsoft Visual C++ .Net Aprenda Ya, Julian Templeman y Andy Olsen.
- Programación con MFC 6.0, Herbert Schildt.

Sitios Web

- www.webopedia.com
- www.whatis.com
- www.about.com
- <http://www.vtmidi.org/>
- <http://www.synthzone.com/>
- <http://www.codeguru.com/>
- <http://www.physik.rwth-aachen.de/~harm/aixphysik/sound/midi/pages/whatmidi.html>
- <http://www.midi.org/>
- <http://www.borg.com/~jglatt/tech/miditech.htm>
- <http://www.programmersheaven.com/>
- <http://www.midimountain.com/>

GLOSARIO DE TERMINOS

ACTIVEX: Es una consecuencia de otras dos tecnologías de Microsoft llamadas OLE (Object Linking Embedding) y COM (Component Object Model). Conjunto de tecnologías que permiten a un componente de software interactuar con otro en un entorno de red, sin importar el lenguaje en el cual hayan sido creados.

ANSI: American National Standards Institute o Instituto Nacional Americano de Estándares, Una organización dedicada al desarrollo de los estándares de la industria americana.

APPLICATION PROGRAM INTERFACE (API): Abreviatura del Application Program Interface, de un conjunto de rutinas, de protocolos, y de herramientas para construir aplicaciones de software. Un buen API hace más fácil desarrollar un programa proporcionando todos los bloques para la construcción de dicho programa. Un programador pone los bloques juntos. La mayoría de los ambientes operativos, tales como Ms-Windows, proporcionan un API de modo que los programadores puedan escribir aplicaciones consistentes con el ambiente operativo. Aunque las APIs se diseñan para los programadores, son en última instancia son buenos para los usuarios porque garantizan que todos los programas que usan un API común tendrán interfaces similares. Esto hace más fácil para que los usuarios aprendan nuevos programas.

ARREGLO O TRACK: Está constituido por varias Pistas y forma una unidad que usualmente contiene una obra (canción, tema, etc.). Será guardado como una Secuencia y de hecho constituye un MidiFile.

BUFFER: Un área de almacenamiento temporal de datos.

BIT: Unidad mínima de información que una computadora puede procesar. Solo puede representar dos valores: 1 y 0 (encendido y apagado).

BYTE: Agrupación de 8 bits.

BYTE DE ESTADO: MIDI-Byte que especifica el comando ó instrucción a ejecutar

BYTE DE INFORMACIÓN: MIDI-Byte que asigna el valor requerido (data) por el Byte de Estado al cual está asociado.

CAMBIO DE PROGRAMA: Mensaje de Voz (de Canal) por el que se accede a un nuevo Programa.

CAMBIO POR CONTROLADOR: Mensaje de Voz (de Canal) por el que se asigna un valor a un Controlador.

CANAL BÁSICO: El Canal Básico es en el cual el instrumento está funcionando (suele tener una asignación de 17 a 32). También llamado Canal Global.

CANAL DIRECCIONADO: Asignación efectuada en el Secuenciador por el cual se ejecutarán los eventos contenidos en la Pista o Patrón (independientemente del Canal por el que se recibieron en su momento).

CANAL MIDI: El MIDI dispone de 16 Canales de comunicación independientes (por Puerto).

CANCIÓN (SONG): En un Secuenciador es la estructura más amplia de que disponemos y está formada por diversos Arreglos.

CARTA DE IMPLEMENTACIÓN MIDI: Tabla con formato estándar (y obligatoria) en la que se especifican las prestaciones MIDI (básicas) del Instrumento.

CHORUS: Efecto que posibilita aumentar la profundidad de la señal tratada logrando una mayor impresión musical. La señal se divide en tres y a cada una de éstas se le aplica un retardo y una panorámica diferente, además de una ligera variación de tono. En los GM, la intensidad del Chorus se maneja con el Controlador #93.

COMPILADOR: Un recopilador es un programa especial que procesa las declaraciones escritas en un lenguaje de programación particular y los convierte en lenguaje de maquina o "código" que el procesador de una computadora utiliza. Típicamente, un programador escribe declaraciones en un lenguaje tal como PASCAL o C en una línea a la vez que usando un editor. El archivo que se crea contiene lo que se llama código fuente. El programador luego ejecuta el compilador apropiado del lenguaje, especificando el nombre del archivo que contiene el código fuente del programa.

Al ejecutarse, el compilador primero analiza todas las declaraciones del lenguaje sintácticamente una después de la otra y entonces, en unas o más etapas o "pasos sucesivos", construye el código de salida, cerciorándose de que las declaraciones que refieren a otras declaraciones están referidas correctamente en el código final. Tradicionalmente, la salida de la compilación se ha llamado código de objeto o a veces un módulo de objeto. El código de objeto es el código automático que el procesador puede procesar o "ejecute" una instrucción a la vez.

Más recientemente, el lenguaje de programación de Java, un lenguaje usado en la programación orientada a objeto, ha introducido la posibilidad de compilar la salida (llamada bytecode) que puede funcionar en cualquier plataforma la cual proporcionen una máquina virtual de Java o un intérprete del bytecode que traduzca las instrucciones del bytecode para que puedan ser ejecutadas por el procesador real del hardware.

Tradicionalmente en algunos sistemas operativos, un paso adicional fue requerido después de la compilación que de resolver la localización relativa de instrucciones y de datos cuando más de un módulo de objeto debía ser ejecutado al mismo tiempo y crear referencias cruzadas a la secuencia de instrucciones de cada uno o a los datos. Este proceso fue llamado a veces edición de enlace y la salida conocida como módulo cargable.

CONTROL ACTIVEX: Control usando tecnología ActiveX. Un control ActiveX puede ser automáticamente descargado y ejecutado en un navegador web. ActiveX no es un lenguaje de programación, sino un conjunto de reglas de cómo las aplicaciones deberían compartir información. Estos controles pueden ser desarrollados en varios lenguajes, incluyendo C/C++, Visual Basic y Java.

Un control ActiveX es similar a un Java applet; pero a diferencia de estos, los controles ActiveX tienen acceso completo al sistema operativo Windows. Esto les da mucho mas poder que los Java applet, pero con un cierto riesgo de dañar programas o datos en la maquina. Para controlar este riesgo, Microsoft ha desarrollado un sistema de registro para que los navegadores puedan identificar y autenticar un control ActiveX antes de descargarlo.

Los controles ActiveX solo pueden ser ejecutados en entornos Windows.

ENCAPSULAMIENTO: Mediante esta técnica conseguiremos que cada clase sea una caja negra, de tal manera que los objetos de esa clase se puedan manipular como unidades básicas. Los detalles de la implementación se encuentran dentro de la clase, mientras que desde el exterior, un objeto será simplemente una entidad que responde a una serie de mensajes públicos.

ENTORNO INTEGRADO DE DESARROLLO: Un entorno integrado de desarrollo (IDE) es un entorno de programación que se ha empaquetado como programa de uso, consistiendo en típicamente un editor del código, un compilador, un depurador, y un constructor del entorno gráfico de usuario (GUI). El IDE puede ser un uso independiente o puede ser incluido como parte de unos o más usos existentes y compatibles. El lenguaje de programación BASIC, por ejemplo, se puede utilizar dentro de Microsoft Office, que permite escribir un programa de WordBasic dentro Microsoft Word. Los IDEs proporcionan un marco amigable para el usuario de uso fácil para muchos lenguajes de programación modernos, tales como Visual Basic, Java, Delphi, etc.

EVENTO MIDI: Acción ejecutada sobre un instrumento MIDI y que genera un Mensaje MIDI.

FORMULARIO: Una colección de unas o más páginas que se comporta como una ventana estándar o una caja de diálogo. Una forma contiene los controles para exhibir y corregir datos. Las formas se pueden contener dentro de un sistema de la forma.

Una parte de la pantalla que puede contener su propio documento o mensaje. En programas basados en ventanas, la pantalla puede dividirse en varias ventanas, cada una de las cuales tiene sus propios límites y puede contener un documento diferente (o una presentación distinta del mismo documento). Cada ventana puede contener su propio menú u otros controles, y el usuario puede ampliarla o reducirla mediante un dispositivo señalador (puntero), que se acciona con el ratón o *mouse*.

INTERNET: La internetwork de redes más grande del mundo, que conecta decenas de miles de redes de todo el mundo y con una cultura que se concentra en la investigación y estandarización basada en el uso real. Muchas tecnologías de avanzada provienen de la comunidad de la Internet. La Internet evolucionó en parte de ARPANET. En un determinado momento se la llamó Internet DARPA, y no debe confundirse con el término general Internet.

LENGUAJE DE PROGRAMACIÓN: Cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por una computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente es la computadora la que realiza la traducción.

MÉTODOS: Una acción que un objeto es capaz de ejecutar.

MENSAJE MIDI: Conjunto de instrucciones y datos asociados con los que se comunican los instrumentos.

MIDI: Lenguaje músico-digital. Acrónimo de Interfaz Digital para Instrumentos Musicales.

MIDI IN: Puerto receptor de Mensajes MIDI.

MIDI OUT: Puerto por el que se envían los Mensajes generados en el propio instrumento.

MICROSOFT WINDOWS: Una familia de sistemas operativos para computadoras personales. Windows domina el mundo de la computadora personal, funcionamiento, por algunas estimaciones, en el 90% de todos los computadores personales. Los 10% restantes son sobre todo computadoras de Macintosh. Como el ambiente de funcionamiento de Macintosh, Windows proporciona una interfaz gráfica de usuario (GUI), gestión de la memoria virtual, tareas múltiples, y la ayuda para muchos dispositivos periféricos.

MODULO: En software, un módulo es una parte de un programa. Los programas se componen de unos o más módulos independientemente desarrollados que no se

combinan hasta que el programa es enlazado. Un solo módulo puede contener una o varias rutinas.

MODOS MIDI: Modo 1: Omni On, Poly On. Modo 2: Omni On, Poly Off. Modo 3: Omni Off, Poly On. Modo 4: Omni Off, Poly Off.

NOTE OFF: Instrucción de Nota Desactivada. Sus valores representan la Velocidad con la que se abandona la Nota: 0-127

NOTE ON: Instrucción de Nota Activada. Sus valores representan la Velocidad con la que se ataca la Nota: 1-127 (Un valor 0 de Note On se traduce en NoteOff ver: Running Status)

MULTIPLE DOCUMENT INTERFACE (MDI): La interfaz múltiple de documento es una interfaz de programación de Microsoft Windows para crear una aplicación que permite a los usuarios trabajar con múltiples documentos al mismo tiempo. Cada documento está en un espacio separado con sus propios controles. El usuario puede ver y trabajar con diversos documentos tales como una hoja electrónica, un documento del texto, o un espacio de dibujo simplemente moviendo el cursor de un espacio a otro.

Una aplicación MDI es algo como la interfaz de escritorio de Windows puesto que ambos incluyen espacios múltiples de visualización. Sin embargo, las vistas del MDI se confinan al área de la ventana o del cliente de la aplicación. Dentro del área del cliente, cada documento se exhibe dentro de una ventana "hija" separada. Las aplicaciones MDI se pueden utilizar para una variedad de propósitos - por ejemplo, trabajar en un documento mientras que lo que se refiere a otro documento, ver diversas presentaciones de la misma información, viendo sitios múltiples en el web, y cualquier tarea que requiera puntos de referencia múltiples y áreas de trabajo en el mismo tiempo.

P-CODE: Es un paso intermedio entre las instrucciones de alto nivel de un programa en Basic y el código nativo de bajo nivel que el procesador de una computadora ejecuta (Visual Basic no genera código de maquina como los compiladores). En tiempo de ejecución, Visual Basic traduce cada declaración del p-code a código nativo. Compilando directamente al formato del código nativo, se elimina el paso intermedio del p-code.

Básicamente, el pcode es un conjunto compilado de instrucciones escritas de tal forma que el ambiente en tiempo de ejecución de Visual Basic pueda traducir las instrucciones para hacer funcionar un programa.

PROPIEDADES: Atributo de un control que define una de las características de u objeto o un aspecto de su comportamiento.

PROTOCOLO MIDI: Conjunto de normas y sintaxis que el Instrumento MIDI debe cumplir para su perfecto funcionamiento.

PUERTO: Conector por el cual se transmiten/reciben datos. p.ej. de impresora, de modem, de joystick, etc. Los Puertos MIDI son: MIDI IN, MIDI OUT y MIDI THRU.

RUNNING STATUS: Norma por la cual se establece que: Mientras los Mensajes consecutivos compartan el mismo Byte de Estado, los siguientes Mensajes (a partir del primero), serán interpretados por el mero hecho de transmitir los Bytes de Información pertinentes. Por consenso y para sacarle máximo partido al Running Status: El valor 0 de Note On equivale a Nota Off.

ANEXOS

ANEXOS

1. Manual de Usuario

Acerca de este manual

Objetivos de este documento

Esta publicación provee la información necesaria que le ayudará a utilizar correctamente MIDIEVENT y así obtener el mejor provecho de esta herramienta, poner en práctica y mejorar los conocimientos teóricos de la música y Tecnología MIDI, prepararte para la realización de tus propias canciones musicales, poner a prueba sus conocimientos musicales, probar y construir melodías musicales a un bajo costo.

Dirigido

Esta publicación está dirigida para dar el soporte necesario sobre MIDIEVENT a los profesores de Música, estudiantes de Música o quienes tienen conocimientos sobre música en general.

Organización de este documento

INTRODUCCIÓN	90
¿QUÉ ES MIDIEVENT?	90
REQUERIMIENTOS DE SISTEMA	91
INICIANDO	92
COMO INSTALAR MIDIEVENT	92
COMO INICIAR MIDIEVENT.....	96
CÓMO UTILIZAR MIDIEVENT	98
CONOZCA MIDIEVENT	98
MENÚ PRINCIPAL	98
Menú Archivo	99
Menú Editores	100
Menú ver	100
Menu Configuracion.....	101
Menú Pista	101
Menú Grabar	102
Menú Tocar.....	102
Menú Entrada	103
Menú Salida	103
PANTALLAS	104
PANTALLA DE CAPTURA DE EVENTOS EN FORMATO HEXADECIMAL	105
PANTALLA DE CAPTURA DE EVENTOS EN FORMATO DECIMAL	106
PANTALLA DE CAPTURA DE NOTAS MUSICALES	107
BARRA DE HERRAMIENTA.....	108
CÓMO CAPTURAR EVENTOS MIDI.....	109
CÓMO TOCAR LOS EVENTOS MIDI.....	110
CÓMO AGREGAR NUEVA PISTA	111

CÓMO MOSTRAR LOS EVENTOS MIDI EN FORMATO DECIMAL	112
CÓMO MOSTRAR LOS EVENTOS MIDI EN FORMATO HEXADECIMAL.....	113
CÓMO MOSTRAR LAS NOTAS MUSICALES	114
CÓMO GUARDAR LOS EVENTOS MIDI	115
CÓMO IMPRIMIR LOS EVENTOS MIDI.....	115
CÓMO HABILITAR TRHU EN EL SECUENCIADOR MIDI	116
CÓMO CONFIGURAR EL TEMPO EN EL SECUENCIADOR MIDI.....	116
CÓMO CONFIGURAR SINCRONIZACIÓN INTERNA E EXTERNA EN EL SECUENCIADOR MIDI	117
CÓMO CONFIGURAR LA RESOLUCIÓN EN EL SECUENCIADOR MIDI	117
CÓMO CONFIGURAR PERIODOS DE TIEMPO EN EL SECUENCIADOR MIDI.....	118
CÓMO CONFIGURAR PERIODOS DE TIEMPO EN EL SECUENCIADOR MIDI.....	118
CÓMO AGREGAR NOTAS EN EL SECUENCIADOR MIDI.....	119