

APLICACION DEL PROTOCOLO MIDI A LA SINTESIS DIGITAL DEL SONIDO

DR. ABELIAN MARINHO FUENTE
FUNDADO COAZEN CIA. LTDA.

DR. HENRI EDUARDO GARCIA
ESCUELA POLITECNICA NACIONAL

RESUMEN

Este trabajo presenta la descripción de dos programas didácticos desarrollados para el estudio del protocolo de comunicación MIDI. El protocolo MIDI es ampliamente utilizado en sistemas de síntesis digital del sonido y es, sin lugar a dudas, el desarrollo tecnológico que más impacto ha tenido en la industria musical. Actualmente ofrece un horizonte de posibilidades ilimitadas de investigación, en el campo de la informática musical.

Luego de una breve descripción del protocolo MIDI, se incluyen los detalles fundamentales del software desarrollado en lenguaje "C" y Assembly. Las pruebas correspondientes se realizaron con la ayuda de un computador personal, un interfaz MIDI MPU-401 y un sintetizador de sonido equipado con puertos MIDI. Los resultados obtenidos, demuestran la utilidad de los programas desarrollados para el estudio y la experimentación de la música asistida por computador.

ABSTRACT

This paper presents a description of two didactic programs that were developed to study the communication protocol known as MIDI. The MIDI protocol is widely used in systems with digital sound synthesizers. With no doubt, MIDI is the technological development that has most influenced the music industry. Currently, it offers endless possibilities for research and development in the field of music information technology.

After a brief description of the principles of the MIDI protocol, the details concerned with the software developed in C and Assembly language are included. The tests were performed using a personal computer, an MPU-401 MIDI interface and a synthesizer with MIDI communication ports. The results obtained, illustrate the facilities that the programs developed offer for studying and experimenting with computer assisted music.

INTRODUCCION

Los sistemas MIDI hicieron su aparición en la segunda mitad de la década pasada, es así que en 1981, los fabricantes ROLAND y-OBERHEIM realizan el primer intento por estandarizar la comunicación entre sus sintetizadores, creando el Universal Standard Interface (USI). A partir de este año varios fabricantes adoptaron este estándar. En 1983 tuvo lugar el primer gran encuentro de fabricantes de sintetizadores y para 1984 apareció el denominado "Musical Instruments Digital Interface" (MIDI). La primera demostración pública del sistema MIDI tuvo lugar en 1985, cuando se conectaron dos

sintetizadores, demostrando que se podía controlar la salida de audio desde el teclado de cualquiera.

En 1987 se creó la "International MIDI Association" (IMA), esta, une a los usuarios de este sistema y juntamente con el "Japanese MIDI Standard Committee" (JMISC), periódicamente establecen las nuevas funciones que deben añadirse a la lista de especificaciones. A partir de ese año, los sistemas MIDI han tenido un desarrollo muy intenso y actualmente la mayoría de instrumentos de síntesis de sonido incluyen por lo menos un puerto de comunicación MIDI.

DEFINICIONES Y CARACTERISTICAS DEL SISTEMA MIDI

El sistema MIDI puede definirse desde dos puntos de vista:

- Musicalmente, el sistema MIDI permite la comunicación de dos sintetizadores, en donde uno de ellos indica al otro que tecla fue presionada, con que fuerza y que presión fue ejercida luego de la presión inicial, así, los más pequeños efectos que pueden aplicarse al sonido durante la ejecución de un instrumento, pueden ser enviados como información digital desde el terminal MIDI OUT al terminal MIDI IN de otro sintetizador, pudiendo este reproducir, con gran precisión, la ejecución del sintetizador maestro.
- Técnicamente, se define al sistema MIDI como un protocolo de comunicación digital en el cual la información se transmite serialmente con una velocidad de 31.25 Kbaudios.

Los terminales de comunicación MIDI, son del tipo DIN de 5 pines, la configuración básica consta de los siguientes terminales:

- MIDI OUT: terminal de salida
- MIDI IN: terminal de entrada
- MIDI THRU: terminal de salida, transmite la misma información recibida por MIDI IN.

El protocolo de comunicación MIDI transmite serialmente mensajes en paquetes de 8 bits. Cada paquete lleva además un bit de inicio (0L) y otro de parada (1L). En la Figura N°1, se presenta la configuración básica de una Unidad de Procesamiento MIDI.

La información MIDI puede ser transmitida a través

de 16 canales independientes, es decir que es posible transmitir información de 16 sintetizadores simultáneamente.

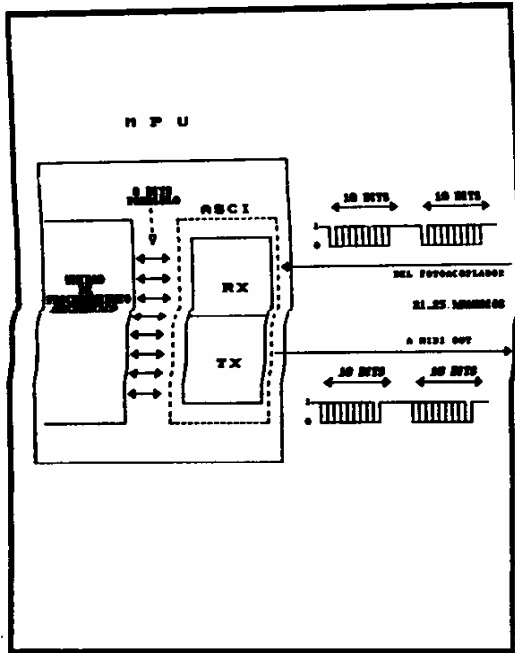


DIAGRAMA DE BLOQUES DE UN EQUIPO DE PROCESAMIENTO MIDI (CPU)

FIGURA Nº 1

FORMATO DE TRANSMISION DE DATOS MIDI

El formato básico de los mensajes MIDI se presenta en la figura Nº 2. En ésta se observa que cualquier mensaje está formado por varios bytes de información, pudiendo haber mensajes que tienen desde un byte hasta 3 bytes de extensión.

El primer byte recibe el nombre de byte de estado, el cual define el tipo de información que será transmitida por los siguientes bytes. El byte de estado siempre tiene como dígito más significativo a 11, en tanto que los bytes que siguen a un byte de estado siempre tienen como byte más significativo a 01, a estos se conoce como byte de datos.

El byte de estado representa dos tipos de información: los 4 bits más significativos representan el tipo de información que será transmitida por los siguientes bytes de datos. Debido a que el bit más significativo en el byte de estado es siempre 11, con los tres bits restantes del nibble más significativo, se definen 8 tipos básicos de mensajes. Los cuatro bits menos significativos representan el canal de transmisión, por lo que se puede definir hasta 16 canales MIDI.

TIPOS DE MENSAJES MIDI

Los mensajes MIDI se clasifican en dos categorías:

- Mensajes de datos de canal.
- Mensajes de datos de sistema

Los mensajes de datos de canal se transmiten siempre a través de uno o más de los 16 canales MIDI, en tanto que los mensajes de datos de sistema se transmiten sin designación de canal.

De manera general, los mensajes de datos de canal representan:

- acciones ejercidas sobre las teclas del sintetizador,
- cambios de sonidos, y
- acciones en los pedales de efectos.

En tanto que los mensajes de datos de sistema transmiten información general como:

- datos de sincronización de reloj en tiempo real,
- mensajes comunes y,
- mensajes exclusivos del sistema.

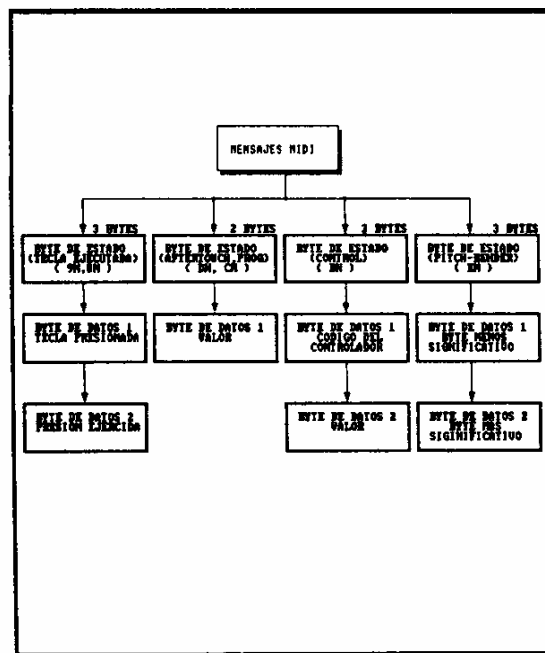


DIAGRAMA DE BLOQUES DE UN MENSAJE MIDI BASICO

FIGURA Nº 2

Existen dos clases de mensajes MIDI de datos de canal:

- mensajes de datos de canal de voz, y
- mensajes de datos de canal de modo.

Los primeros son los más comunes mensajes MIDI, estos transmiten 7 tipos de información diferentes, la que permite el control del sistema de síntesis de sonido, estos mensajes corresponden a códigos de: tecla presionada, tecla liberada, velocidad, modulador de frecuencia (pitch bender), presión posterior (after-touch), cambio de programas (sonidos) y

cambio de controladores.

En tanto, los mensajes de datos de canal de modo envían información para que un sintetizador esclavo pueda interpretar la información recibida desde un sintetizador maestro.

MENSAJES DE DATOS DE CANAL DE VOZ

Son transmitidos por cualquiera de los 16 canales y son recibidos solo por un usuario que esté asignado al mismo número de canal. Estos mensajes tienen por lo menos 2 bytes, el primer byte corresponde al byte de estado y los siguientes a los bytes de datos.

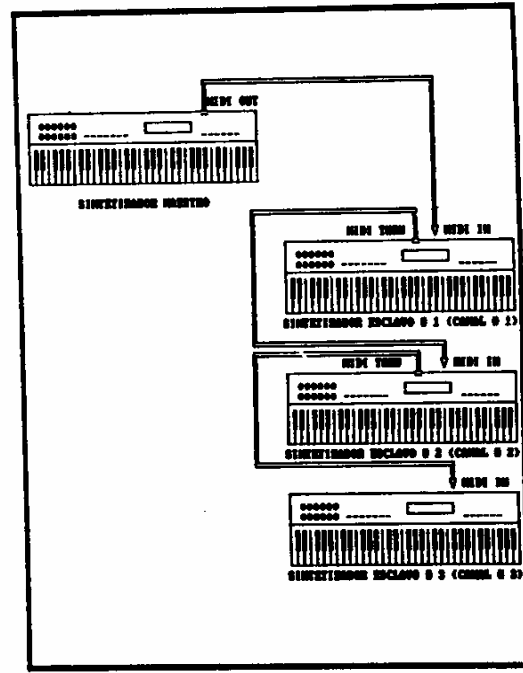
Mensaje de nota de tecla presionada

Sirve para transmitir información de la tecla ejecutada en el sintetizador. Está formado por 2 bytes:

- El primero es el byte de estado y tiene la forma 1001mmn (9nH), donde el bit más significativo es 1L y al nibble menos significativo corresponde al número de canal MIDI de transmisión.
- El segundo byte corresponde a la nota presionada. Por ser un byte de datos, tiene el bit más significativo igual a 0L. Los 7 bits restantes sirven para identificar 128 notas musicales. Para transmitir información de nota presionada, el protocolo MIDI define las notas musicales partiendo desde la nota DO de la octava -2 hasta el SOL de la octava 8, (total 128 notas musicales). Para el teclado MIDI estándar las notas musicales empiezan en la octava 1 y van numeradas, en hexadecimal, desde la 24H hasta la 60H. La nota DO central (262 Hz), corresponde al código 3Ch. (Figura Nº 4)

Mensaje de nota de tecla liberada

Este mensaje es muy similar al anterior. Indica la



SINTETIZADORES EN CONDICIÓN DE VOZ CANAL

FIGURA Nº 3

MENSAJES MIDI		MENSAJES DE DATOS DE SISTEMA (no se transmiten por ningún canal)
MENSAJES DE DATOS DE CANAL (transmitidos por canal n)		
CANAL DE VOZ	CANAL DE MODO	<ul style="list-style-type: none"> - Reloj MIDI - Comunes del sistema - Exclusivos del sistema
<ul style="list-style-type: none"> - tecla presionada - tecla liberada - velocidad - presión posterior - cambios de control - pitch bend - cambios de programa. 	<ul style="list-style-type: none"> - Modos de recepción 	

nota de tecla que se ha liberado en el sintetizador. Su extensión es de 2 bytes, correspondiendo al byte de estado al tipo 8nH. El segundo byte, corresponde al código de la tecla liberada.

Mensaje de velocidad ejercida durante la presión

Sirve para codificar la velocidad ejercida en el instante de accionar una tecla en el sintetizador, la que es equivalente a la presión ejercida en la misma. Este mensaje se transmite como un tercer byte de datos en los mensajes de tecla presionada y tecla liberada.

Para la identificación del byte de velocidad, se definen equivalencias entre los 128 valores posibles y su correspondiente musical. (Figura Nº 5).

Mensaje de "after-touch". (Presión posterior)

Se define como presión posterior a la presión aplicada a una tecla una vez que ha finalizado la activación de la misma. El número de bytes correspondientes a este mensaje es de 2, el byte de estado puede ser DnH o AnH (si es after-touch polifónico), donde "n", corresponde al número de uno de los 16 canales de transmisión.

CARTA DE DINAMICA DE VELOCIDAD		
FUERZA	VALOR DE VELOCIDAD MIDI	
ppp (pianisimo)	20	14h
pp (muy piano)	40	28h
p (piano)	50	32h
mp (mezzo piano)	64	40h
mf (mezzo forte)	70	46h
f (forte)	80	50h
ff (muy fuerte)	90	5Ah
fff (fortisimo)	115	73h

Mensajes de cambios de control.

Estos mensajes transmiten información referente a acciones ejercidas sobre los controladores de teclado, es decir elementos adicionales que permiten alterar los sonidos generados por el sistema de síntesis del sintetizador.

En el protocolo MIDI, se han establecido 128 diferentes tipos de controladores de teclado. Hasta la actualidad, no se han desarrollado más de 20 controladores, por lo que la mayoría están todavía indefinidos. Entre los controladores más comunes se puede mencionar: rueda de modulación, filtros de modulación, pedal de expresión y pedal de sostenido.

Una de las características más importantes del sistema MIDI, radica en el hecho de que se permite la asignación indistinta de cualquier controlador a cualquier parámetro del sonido que se desea controlar, esto permite adaptar el sintetizador a los más variados gustos del interprete.

Estos mensajes están constituidos por 3 bytes de información. El byte de estado correspondiente es BnH. El segundo byte identifica el controlador sobre el cual se ejecutará la acción. El tercer byte expresa el valor que será asignado al controlador respectivo.

Mensajes de "Pitch bend".

Este mensaje se usa para transmitir acciones sobre el controlador conocido como "pitch bender". Este permite modular en frecuencia al sonido generado por el sintetizador. Actualmente se lo considera como un controlador más, aunque en las primeras versiones de las especificaciones MIDI se lo trataba individualmente. Consta de 3 bytes, siendo su byte de estado BnH. Los bytes de datos entregan información acerca de sus variaciones.

Mensajes de cambios de programa.

Se define como programa a una serie de ajustes en los parámetros de síntesis del sintetizador. El cambio de sonidos genera mensajes de cambios de programa, estos tienen una extensión de 2 bytes. El byte de estado correspondiente es: CnH. El segundo byte corresponde al número de sonido accionado, por lo que existe la posibilidad de conmutar a 128 sonidos diferentes.

MensAJES DE DATOS DE CANAL DE MODO

Este tipo de mensajes son transmitidos por cualquier canal y llevan información de la manera como un sintetizador esclavo debe recibir la información desde un sintetizador maestro. Estos se transmiten como si fueran mensajes de cambios de control. Están formados por 3 bytes de información, donde el byte de estado es BnH.

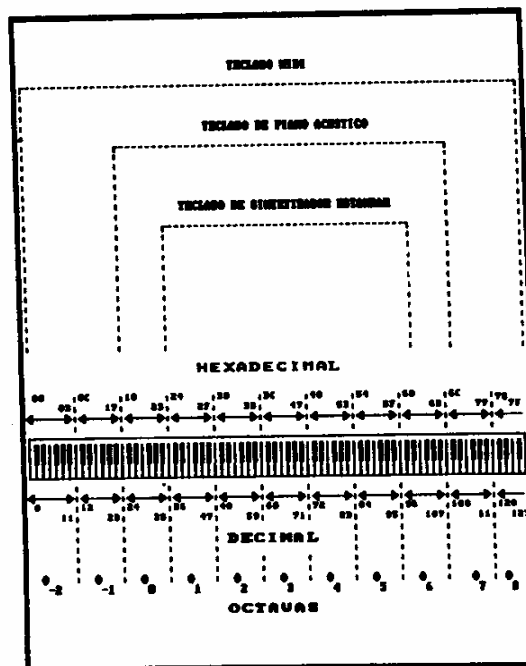
MENS AJES DE DATOS DEL SISTEMA

No llevan asignación de ningún canal y sirven para

transmitir información del sistema MIDI. Los mensajes de datos de sistema son de tres tipos: mensajes de reloj en tiempo real, mensajes comunes del sistema y mensajes exclusivos del sistema.

Reloj MIDI.

Debido a que toda acción de un sistema MIDI es en tiempo real, se usa un reloj que permite sincronizar al mismo. Este es muy importante en aplicaciones de grabadoras digitales MIDI (secuenciadores). Todo elemento MIDI, permite la selección de un reloj ya sea interno o externo, para un adecuado funcionamiento de un sistema MIDI, todos sus aparatos deben estar asignados a un mismo reloj. Los mensajes de reloj MIDI son todos de un byte de información, siendo el más importante el correspondiente al valor FEh, el cual se transmite cada 50 ms.



CONDICIONES DECIMALES Y HEXADECIMALES DEL TECLADO DEFINIDAS POR EL PROTOCOLO MIDI

FIGURA Nº 4

Mensajes comunes del sistema.

Son usados para enviar información del tiempo musical así como para llevar control de un sintetizador analógico conectado al sistema.

Mensajes exclusivos del sistema

Son usados para transmitir información almacenada en la memoria del sintetizador, la misma que contiene los ajustes de los diferentes elementos de síntesis. Este tipo de mensajes son especialmente útiles para transferir datos de la memoria de un sintetizador a un computador para ser procesados por el mismo y posteriormente devueltos al sintetizador.

IMPLEMENTACION DE LOS PROGRAMAS

Actualmente, las computadoras son uno de los elementos más importantes de un sistema MIDI, en el presente trabajo se han diseñado e implementado dos programas que permiten visualizar con claridad dos posibilidades de este tipo de sistema: un analizador de información MIDI y un secuenciador digital MIDI.

Los lenguajes escogidos para la implementación de los programas son el "C" y el Assembly. La velocidad de procesamiento es el factor determinante al escoger estos lenguajes. Las subrutinas de comunicación han sido escritas en lenguaje Assembly, en tanto que el lenguaje C ha sido usado para la decodificación de los mensajes MIDI, así como para crear los diferentes interfaces al usuario.

El hardware usado para la implementación de los programas consta de los siguientes elementos: Interfaz (V-4000 de VOYETRA) compatible con la Unidad de Procesamiento MIDI de ROLAND MPU-401, que es el estándar MIDI para el IBM-PC, sintetizador equipado con por lo menos un puerto de comunicación MIDI, y un computador IBM-AT.

El interfaz ROLAND MPU-401, contiene al microprocesador ROLAND 6801, el cual tiene internamente programado las secuencias de sincronización de la información MIDI, así como el control de una salida de audio que permite obtener una señal de metrónomo de la información musical. Los puertos de comunicación de este interfaz con el computador corresponden a: 330H para datos y 331H para comandos de control.

- PROGRAMA ANALIZADOR

El Analizador de datos MIDI realiza la decodificación de la información recibida desde el sintetizador en sus equivalentes musicales, de tal manera que permite visualizar los códigos hexadecimal y la respuesta a estos al ser transmitidos al sintetizador, desde el computador.

Este programa, reconoce los códigos MIDI recibidos, de tal manera que a cada posible mensaje MIDI asocia su equivalente musical, desplegando esta información en pantalla. Debido a su naturaleza didáctica y al manejo de ayudas sobre el protocolo MIDI, el programa analizador es una herramienta muy útil de aprendizaje para desarrollar aplicaciones MIDI.

Para establecer la comunicación sintetizador - computador, se han desarrollado en lenguaje Assembly, tres subrutinas denominadas: "recibir dato", "enviar dato", "enviar comando". Como su nombre sugiere, estas subrutinas básicamente permiten recibir y transmitir información codificada de los diferentes comandos de control entre el MPU-401 y el computador.

Estas subrutinas están escritas en tres módulos separados y definidas como funciones públicas, las que posteriormente son incorporados por el "linker" con los módulos escritos en lenguaje C.

El diagrama de flujo básico se presenta en la figura N° 5. La información que contiene el MPU es presentada al computador en el puerto de datos número 330H. El código que presenta el MPU para informar al computador de la presencia de un dato en el puerto de datos corresponde al comando: "Data Send Ready" (DSR), cuyo equivalente es (01111111) (0 lógico en el bit # 7), este código es enviado por el puerto de estado signado con el número 331H. Es decir que si se comprueba en el puerto número 331H, un byte con su bit más significativo en 0 lógico, el MPU tiene un byte de dato en el puerto número 330H, listo para ser enviado al computador. Las subrutinas de comunicación prevén además un lapso de tiempo para leer el dato en el puerto de datos, esto es útil ya que no siempre puede darse la condición de que el MPU esté listo a enviar un dato al instante que el computador solicita el mismo. Para este fin se incluye un lazo que se repite un cierto número de veces, de tal manera que en este tiempo aparezca un comando de "Data send Ready" en el puerto 331H. Si no se incluyera este lazo y si por ejemplo se hubiese desconectado algún cable entre el sintetizador y el computador, este se quedaría esperando el dato indefinidamente. Si en el intervalo de tiempo dado por este lazo, el MPU no responde, la subrutina devuelve en lugar de dato un mensaje de error que es definido con: -1. Esto informa al programa escrito

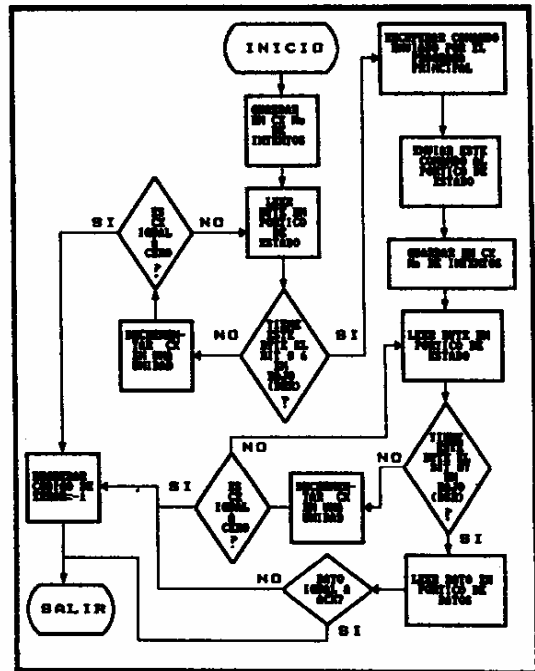


DIAGRAMA DE FLUJO DE LA RUTINA EN ASSEMBLY "COMANDO"

FIGURA N° 5

en lenguaje "C", que no existe dato en el p6rtico de datos del MPU o que existe alg6n problema de conexiones entre el computador y el sintetizador.

El programa ANALIZER, basa su funcionamiento en las subrutinas de comunicaci6n escritas en lenguaje Assembly. La l6gica para la recepci6n y transmisi6n de la informaci6n se basa en el diagrama de flujo de la figura N6 6, donde existe un lazo infinito en el cual se realiza la decodificaci6n, recepci6n y transmisi6n de la informaci6n. Para decodificar cada posible mensaje MIDI se usan funciones independientes las cuales son llamadas desde un m6dulo principal, el mismo que permite el manejo del interfaz al usuario, subrutinas de comunicaci6n como funciones publicas, ayudas y m6dulos de decodificaci6n. Estos m6dulos han sido escritos en lenguaje C. Debido a que por la velocidad de transmisi6n y por la cantidad de datos procesados, el computador no puede decodificar los datos de manera inmediata, el programa permite el manejo de un buffer temporal de datos.

El interfaz que el programa analizador presenta al usuario, se muestra en las figuras N6 8 y 9, en esta 6ltima se presenta una de las varias pantallas de ayuda del protocolo MIDI.

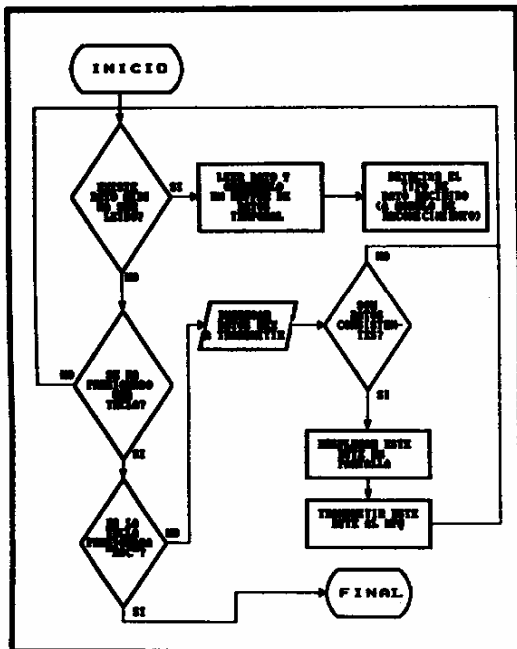


DIAGRAMA DE FLUJO DEL PROGRAMA ANALIZER (PROGRAMA PRINCIPAL)

FIGURA N6 6

- PROGRAMA SECUENCIADOR

Los secuenciadores basados en software son una de las m6s comunes aplicaciones de los computadores a los sistemas MIDI. Se conoce como secuenciador a un sistema que permite el registro en tiempo real de informaci6n MIDI, la cual puede ser grabada y reproducida de manera similar a los sistemas de grabaci6n multipista anal6gicos, es decir en pistas polif6nicas independientes. De esta manera la

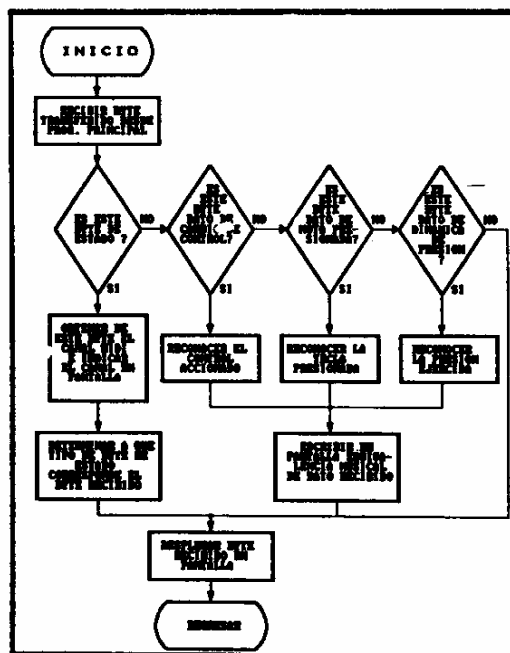


DIAGRAMA DE FLUJO DE RUTINA DE DETECCIÓN DE MENSAJE RECIBIDO

FIGURA N6 7

informaci6n puede ser procesada de m6ltiples maneras, entre las que se menciona:

- alteraci6n de la duraci6n,
- transposici6n,
- asignaci6n de cualquier parte de la secuencia a cualquier sonido, etc.

Una de las principales caracter6sticas de los secuenciadores MIDI, radica en el hecho de que el espacio de memoria requerido es muy compacto. Como comparaci6n, una secuencia musical bastante elaborada al ser grabada con un secuenciador MIDI, no requerir6 m6s de 30 o 40 Kbytes, en tanto que el espacio requerido por la misma secuencia al ser registrada con audio digital de alta calidad ser6 en el orden de los Mbytes.

El programa secuenciador dise1ado en el presente trabajo, permite el registro y reproducci6n de la informaci6n MIDI en una pista polif6nica, asign6ndose a la secuencia el tiempo musical y velocidad que se desee. El metr6nomo, puede ser activado y desactivado por medio de acciones en el men6 del programa, as6 como permite listar las secuencias registradas a las que les a1ade autom6ticamente la extensi6n .MID. El secuenciador, presenta adem6s una ventana, donde se indica el estado de operaci6n del mismo. El interfaz al usuario ha sido dise1ado usando men6s desplegables de tal manera que emula la presentaci6n de las grabadoras anal6gicas.

La comunicaci6n sintetizador - computador se basa en las mismas subrutinas de comunicaci6n que usa el programa analizador.

El dise1o del programa es modular, existen funciones independientes escritas en lenguaje C, que se

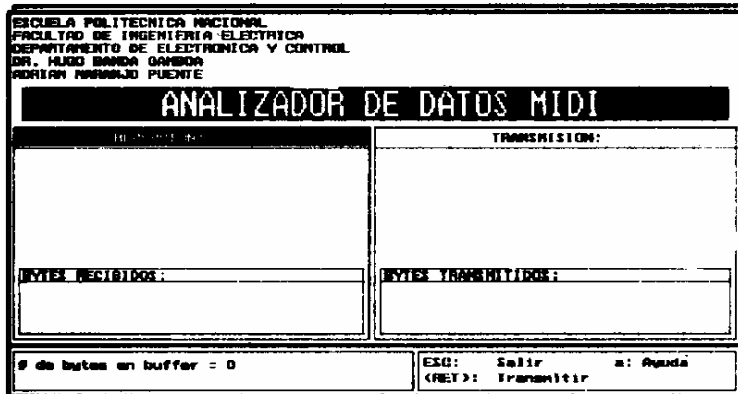


FIGURA Nº 8

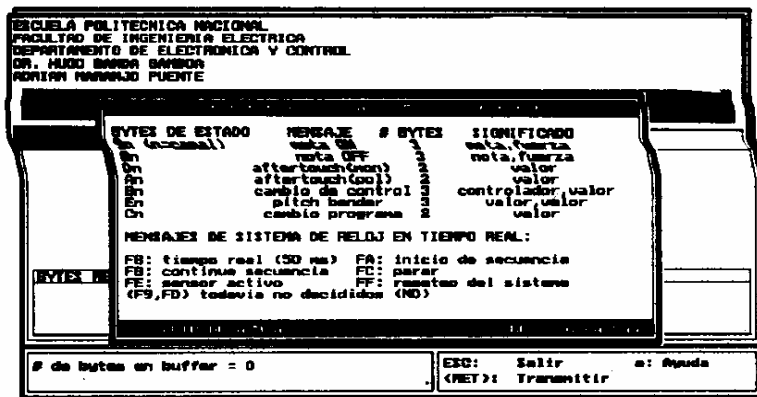


FIGURA Nº 9

encargan de cada procesamiento seleccionado en el menú principal. (figura Nº 10).

La función de grabación GRABAR(), se basa en el diagrama de flujo de la figura Nº 11. Con el uso del comando del MPU-401: COMANDO(START REC), se inicializa al MPU-401 en el modo de grabación. El lazo de grabación de eventos es infinito, el mismo que termina solo si existe acción por parte del usuario. La lógica para determinar la longitud de cada evento es la siguiente: se lee el primer dato transmitido por el MPU-401, dependiendo del valor de este byte se toman dos alternativas. Si el byte leído es menor que EFh, significa que corresponde a un byte de tiempo (ningún dato MIDI es menor a EFh). En este caso se debe leer el byte siguiente, y dependiendo del valor del segundo byte se determina el número de bytes que tendrá el mensaje. Así, si el segundo byte es menor que 7Fh, significa que este byte es de datos. En este caso se debe recuperar el byte de estado de memoria, ya que el MPU-401 no transmite en cada mensaje el byte de estado, este se transmite sólo si el mensaje ha cambiado de byte de estado. El byte de estado ausente en el mensaje, corresponde al byte almacenado en la variable "estado", en la cual se almacena todo byte de estado que registre la función. Para este caso se debe tomar el tercer dato, y se almacenan en memoria con la función GUARDAR(), de la siguiente manera: inicialmente se almacena el byte 11h, el cual corresponde a un indicador de que

las secuencias registradas con este programa corresponden al estándar MIDI 0.1, así el secuenciador reproducirá solamente secuencias que contengan su primer byte igual a 11h, posteriormente, se almacena el número de bytes del evento y los bytes correspondientes al mismo. La comparación del segundo byte con los diferentes mensajes MIDI, se usa para determinar la longitud de todos los posibles eventos.

En el caso de que el primer byte leído no corresponda a un valor menor a EFh, se deduce que este byte corresponde a un código del MPU-401, por lo tanto el evento corresponde solo a un byte, siendo almacenado sólo un byte por eventos con el uso de la función GUARDAR().

En el caso de que no se cumpla ninguna de las condiciones indicadas, el proceso de grabación se detiene indicando un mensaje de error. ("Dato no reconocido")

Este método de decodificación de la información MIDI, es posible gracias a la manera como ha sido definido el protocolo. Así, los bytes más comunes tienen siempre valores menores, lo que permite usar una sencilla lógica de decodificación de los mismos.

La función GUARDAR(), recibe como argumento los datos de cada evento, definiendo un archivo cuyo

ESCUELA POLITECNICA NACIONAL
 FACULTAD DE INGENIERIA ELECTRICA
 DEPARTAMENTO DE ELECTRONICA Y CONTROL
 DR. HUGO BANDA GAMBOA
 ADRIAN NARANJO PUENTE
 PROGRAMA: SECUENCIADOR MIDI PARA EL
 INTERFAZ ROLAND MPU - 401

BPM	Ajustar el numero de beats del metrónomo	:	
TIEMPO	Ajustar el # de beats por compás.	:	/4
METRONOMO	Apagar / encender el metrónomo	:	
GRABAR	Inicio de grabación		
EJECUTAR	Inicio de ejecución		
LISTAR	Lista archivos .MID		
SALIR	Regresar al DOS		

mentos, al ser grabada con el secuenciador desarrollado ha requerido aproximadamente 15 Kbytes de espacio en memoria. Si esta pieza fuese registrada con algún sistema de audio digital requeriría seguramente más de 5 Mbytes de memoria.

Es importante mencionar que si bien el espacio para registrar la información MIDI es bastante pequeño, la reproducción de esta información, requiere necesariamente un sintetizador MIDI, para que transforme los códigos respectivos en señales de audio.

Debido a que el potencial del protocolo de comunicación MIDI es bastante extenso, los programas diseñados han permitido visualizar sólo algunas de las posibilidades básicas. Como recomendaciones se presentan algunas de las posibles alteraciones a los mismos:

- El envío de la información desde el computador hacia el sintetizador, se basa en el ingreso de los códigos MIDI por el usuario, una posibilidad más interesante sería, ingresar la información con notación musical. De esta manera, se desplegaría en pantalla una escala musical y se escogería en esta los parámetros musicales a transmitirse.
- Debido a la existencia de mensajes MIDI exclusivos, es posible la transmisión de información de parámetros de sonidos almacenados en la memoria del sintetizador. El programa analizador, puede ser usado como una etapa inicial de un sistema que permita decodificar estos datos y almacenarlos en memoria, para así realizar alteraciones a los mismos y así poder transmitirlos al sintetizador. Así, sería posible la síntesis directa sobre los sonidos del mismo.
- La implementación de un secuenciador multipista, se facilita debido a que el sistema de almacenamiento de datos MIDI es a nivel de eventos. Así, si a cada evento se le añade un byte donde se indique la pista a la cual está asignado, es posible enrutar cada evento a pistas diferentes. Esta alteración

ampliaría enormemente las posibilidades del secuenciador.

- Los mensajes de fin de compás que transmite el MPU, pueden ser usados de tal manera de marcar en pantalla los tiempos del compás elegido al iniciar el metrónomo.
 - Al ser detenida la ejecución de una secuencia en el sintetizador, el programa secuenciador regresa automáticamente al inicio de secuencia. Esta situación puede ser alterada si se mantiene un registro del número de eventos y sus extensiones, de esta manera es posible emular las funciones de avance rápido y rebobinado de grabadoras analógicas. Esto permitiría ejecutar una secuencia en cualquier punto de la misma.
 - Existen varias alternativas para editar la información musical incluida en una secuencia. Por ejemplo, se recomienda la implementación de un sistema de transposición de altura tonal. Esto se podría lograr guardando registro de los eventos MIDI que corresponden a nota presionada y liberada, de tal manera que se sume o reste al segundo byte del mensaje u valor ingresado por el usuario que debe ser correspondiente al valor de la transposición.
 - Una posible extensión del programa secuenciador, constituye la transformación de los códigos MIDI en equivalentes musicales para ser graficados en notación musical (partitura). Esto puede ser logrado usando las marcas de compás y mensajes de reloj del MPU-401 y los bytes de datos de nota presionada y liberada.
- En general para realizar la edición de cualquiera de los posibles eventos MIDI, se debe guardar registro de todos los eventos de un mismo tipo, para así modificar los bytes de datos respectivos.

BIBLIOGRAFIA

- BORLAND. TURBO C USER'S GUIDE, Ver. 1.5. 1988.
- CASIO Corporation. CASIO MT-700 DIGITAL SYNTHESIZER USER'S GUIDE. Japón, 1989.
- Conger, Jim. C PROGRAMMING FOR MIDI. 1ra. ed. NY. RAT Publishing, Inc, 1990.
- Chamberlain, H. MUSICAL APPLICATIONS OF MICROPROCESSORS. NY. Hayden Books, 1985.
- Deutah, Herbert. SOUND SYNTHESIS. NY. Tab Books, 1983.
- Holmes, Thomas. ELECTRONIC AND EXPERIMENTAL MUSIC. NY. Charles Scribener's Sons, 1987.
- Letraublion, G. MUSICA ELECTRONICA. 2da. ed. Madrid. Editorial Paraninfo, 1982.
- Nathava, Max. THE TECHNOLOGY OF COMPUTER MUSIC. NY. Harcourt Books, 1984.
- Neelsh, Soma y Shama. Neelsh. ADVANCED TURBO C PROGRAMMER'S GUIDE. NY, John Wiley & Son. Inc, 1989.
- Norton, Peter. ASSEMBLY LANGUAGE BOOK FOR THE IBM PC. NY. Prentice Hall, 1986.
- ROLAND Corporation. FUNDAMENTOS DE MUSICA ELECTRONICA CON SINTETIZADORES. Osaka. Japón. 1983.
- ROLAND Corporation. MIDI PROCESSING UNIT MPU-401 TECHNICAL REFERENCE MANUAL. Japon, 1989.
- Scanlon, Leo. 8086 / 8088 / 80286 ASSEMBLY LANGUAGE. NY. Sison & Schuster, 1988.
- VOYTRA Technologies. V-400X USER'S GUIDE. NY. 1992.
- Walker, Don. NOW MIDI WORKS. CA. Peter Alexander Publishing. Inc. 1992.
- YAMAHA Corporation. WHAT'S MIDI. Japón, 1990.

BIOGRAFIAS

ING. ADRIAN NARANJO PUNTE



Nació en Quito el 20 de Agosto de 1964. Se graduó de Bachiller FFMM en el Colegio San José "La Salle" de Guayaquil. Obtuvo el Título de Ingeniero en Electrónica y Control en la Escuela Politécnica Nacional en el año de 1993.

Actualmente presta sus servicios como Representante Técnico de la compañía Proteco Coasin Cia. Ltda.

DR. HUGO BANDA GAMBOA.



Nació en Quito el 7 de Julio de 1950. Sus estudios secundarios los realizó en el Colegio Central Técnico, obteniendo el título de Bachiller Técnico en Radiotecnía. En la Escuela Politécnica Nacional realizó sus estudios superiores, graduándose de Ingeniero en Electrónica y Telecomunicaciones, en 1975. La formación de Postgrado la realizó en el Reino Unido de la Gran Bretaña. En 1978, la Universidad de Bradford le confirió el Título de MSc. en "Power Electronics", y en 1990 la Universidad de Dundee, le otorgó el Título de PhD en "Computer Vision". Ha realizado diversos trabajos de investigación en aplicaciones de la electrónica y la informática tanto en el campo industrial, como en el de la bioingeniería. Sus campos de interés comprenden la automatización industrial y las aplicaciones de la inteligencia artificial. Actualmente se desempeña como profesor principal en el Departamento de Electrónica y Control de la Facultad de Ingeniería Eléctrica de la Escuela Politécnica Nacional.