

# II MIDI

Massimiliano Salfi

[Massimiliano.salfi@gmail.com](mailto:Massimiliano.salfi@gmail.com)

# La storia

Dietro la sigla MIDI, acronimo di **M**usic **I**nstrument **D**igital **I**nterface (interfaccia digitale per strumenti musicali) sono contenuti due concetti:

- l'*interfaccia*, cioè l'hardware standard di cui gli strumenti devono essere dotati per comunicare;
- uno *standard di comunicazione* (un protocollo), ossia un insieme di regole e messaggi interpretabili in modo univoco che permettono agli strumenti di scambiarsi informazioni.

I messaggi che gli strumenti si possono scambiare sono svariati: suona il DO della terza ottava, abbassa il volume al pianoforte, manda a destra il suono del violino, ecc. Questi messaggi viaggiano attraverso un cavo che collega gli strumenti.

# La storia

Il protocollo MIDI nasce all'inizio degli anni '80 grazie agli studi di due ingegneri di Sequential Circuits (SCI), D. Smith e C. Wood, che nel 1981 propongono le prime specifiche del MIDI in un documento pubblicato sotto il nome di "The complete SCI MIDI".

Rendere i nuovi strumenti digitali in grado di comunicare e di sincronizzarsi tra loro, era diventata una necessità per i musicisti e per i produttori stessi. Diversi costruttori, ad esempio Oberheim e Roland, offrivano già sui propri strumenti alcuni sistemi di interfacciamento.

Queste interfacce però, basate su algoritmi proprietari, garantivano il proprio funzionamento solo su strumenti dello stesso costruttore. Il protocollo di Smith e Wood si presentava invece come un sistema in grado di superare questo limite.

# La storia

Per garantire la piena compatibilità tra i vari strumenti, al di là della casa produttrice, ogni costruttore fu invitato a partecipare alla stesura definitiva delle prime specifiche MIDI. SCI, Roland, Yamaha e Kawai furono i primi produttori di strumenti digitali ad aderire alla definizione e alla diffusione del MIDI.

Nel 1982 fu presentato ufficialmente il "MIDI 1.0" ossia le specifiche del primo vero standard di interfacciamento tra apparecchiature musicali. Come vedremo in seguito, il progetto iniziale era talmente ben definito da necessitare di pochissimi aggiornamenti nel corso di questi trent'anni.

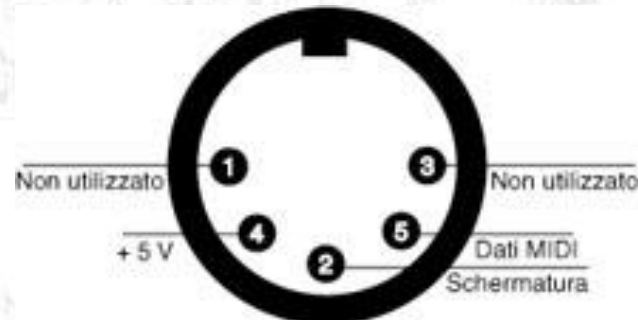
Il primo synth dotato di interfaccia MIDI, fu presentato nel 1983. Si trattava del PROPHET 600 di SCI.

# L'interfaccia hardware

Dal punto di vista circuitale, l'interfaccia MIDI non è altro che un'interfaccia seriale asincrona con "data rate" di 31,250 Kbps, ovvero 31250 "bit per second".

Data la sua natura asincrona, oltre al segnale di massa e di alimentazione da collegare alla scheda remota tramite una resistenza da 220 Ohm, c'è la linea dai dati, che può essere di trasmissione, o di ricezione, a seconda dei casi.

In genere gli strumenti e gli apparecchi musicali sono dotati di 3 connettori di tipo DIN (Deutsche Industrie Normen) a 5 poli la cui pedinatura è illustrata nel seguente schema:



Un connettore DIN a cinque pin

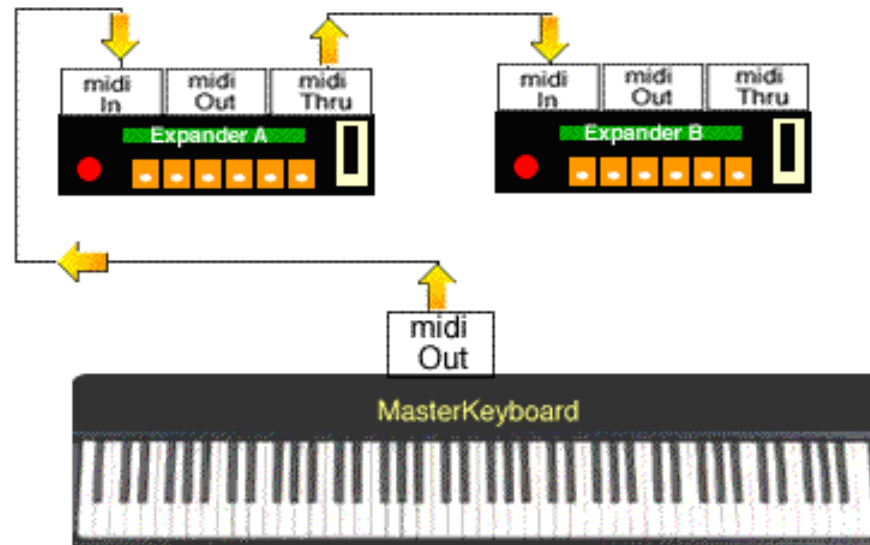
# L'interfaccia hardware

I tre connettori vengono distinti come:

- **MIDI In:** Porta a cui arrivano i dati. Questa porta deve essere sempre collegata con una porta MIDI Out oppure Thru. Non si possono collegare più porte ad un solo In.
- **MIDI Out:** Porta da cui escono i dati verso un MIDI In. Tutto quello che viene eseguito sulla tastiera può essere mandato attraverso questa porta. Non si possono collegare più porte ad un solo Out.
- **MIDI Thru:** Il MIDI Thru è una copia esatta dei dati che arrivano al MIDI In e vengono immediatamente rispediti fuori attraverso il MIDI Thru. Questa porta serve per collegare ad uno stesso master (la macchina che comanda il sistema) più slaves (le macchine a cui arrivano i dati MIDI).

# Esempi di collegamento MIDI

Una possibile configurazione è illustrata nella seguente figura:



I dati escono dal **Midi Out** della tastiera, attraverso la presa **Midi In** vengono ricevuti dall'expander A il quale li reindirizza verso l'expander B attraverso la propria porta **Midi Thru**.

# Esempi di collegamento MIDI

Nella figura seguente è illustrato uno dei collegamenti classici del MIDI. In essa le frecce evidenziano il flusso dei dati.

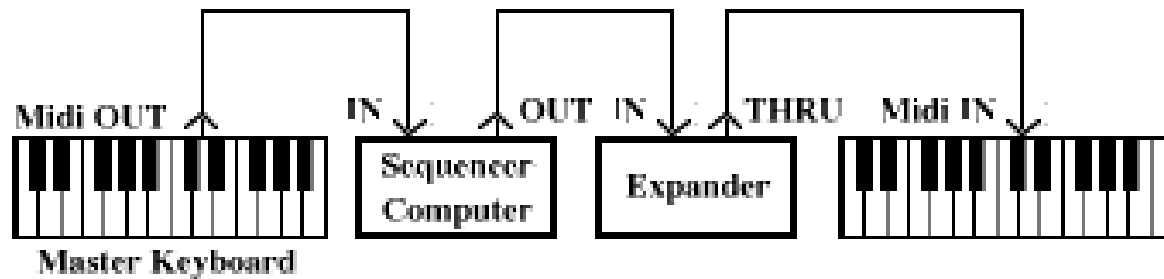


Con questo tipo di collegamento bisogna considerare l'eventuale ritorno MIDI: i dati che entrano nella presa **Midi In** del sequencer riescono dalla presa **Midi Out** creando un riciclo di dati (loop) tale da bloccare il sistema, specialmente se l'apparecchiatura che svolge la funzione di Sequencer è un computer. Ciò deriva dal fatto che l'interfaccia del Computer è quasi sempre priva della presa **Midi Thru** e per default quello che entra nella presa In esce dalla presa Out.

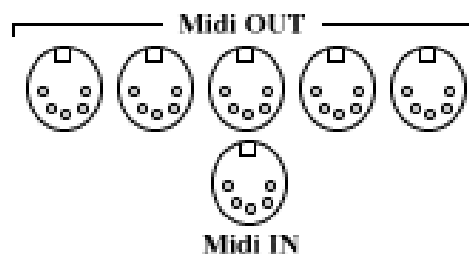


# Esempi di collegamento MIDI

Nella figura seguente è illustrato il collegamento classico utilizzato con una Master Keyboard muta:

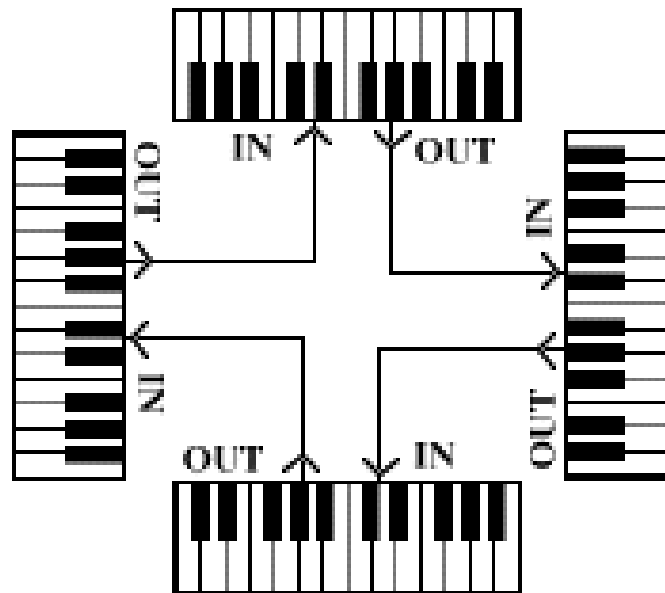


In questo collegamento viene utilizzata la presa **Midi Thru**. A causa della natura seriale del flusso MIDI, la soluzione sopra (in cascata) va limitata in quanto provoca un ritardo fastidioso e inutilizzabile del suono in uscita. La soluzione ottimale per collegare più tastiere o expander è l'utilizzo di Midibox:



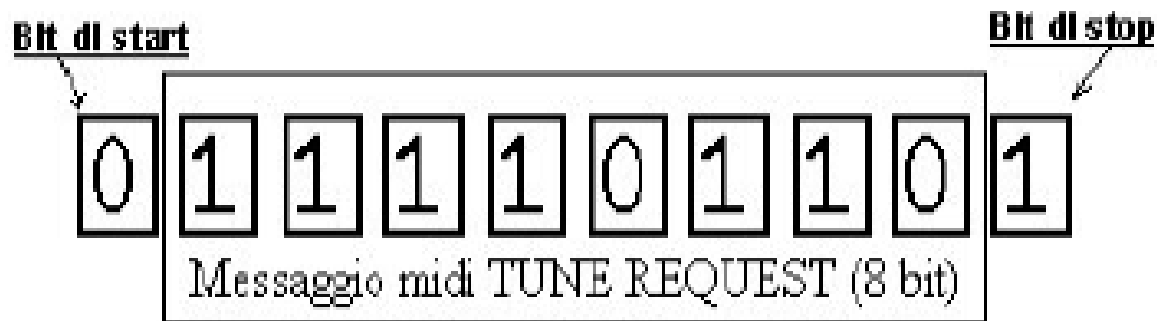
# Esempi di collegamento MIDI

Per tastiere che lo consentono, un'alternativa al collegamento attraverso la presa **Midi Thru**, è il **Collegamento Ring**:



# Il protocollo e la sintassi

Essendo, come detto, la trasmissione asincrona, è necessario disporre di un bit di start e uno di stop, oltre agli otto bit della trasmissione. Quindi una parola che viene ricevuta o trasmessa dall'interfaccia MIDI è composta da 10 bit:



Il sistema MIDI è fondato sull'utilizzo di canali indipendenti (16 per ogni modulo sonoro, di cui uno, il 10, è interamente dedicato alle percussioni) ed è quindi in grado di inviare più suoni contemporaneamente.

# I messaggi MIDI

Un messaggio MIDI è composto da un insieme di byte. Si possono distinguere due tipi di byte:

- status byte (byte di stato);
- data byte (byte di dati).

Un messaggio MIDI deve sempre iniziare con uno status byte. Quindi gli status byte sono quelli inviati per primi e servono a decifrare i data byte successivi, senza possibilità di errori.

# I messaggi MIDI

- Gli status byte servono per definire in modo univoco un comando. Il bit più significativo dello status byte è uguale ad 1 e quindi uno status byte può assumere un valore compreso tra 128 e 255. Gli status byte trasmettono il tipo di informazione (suona una nota, alza il volume, ecc).
- I data byte, invece, servono per inviare gli eventuali parametri necessari per un corretto funzionamento dello status byte. Il bit più significativo del data byte è uguale a 0 e quindi un data byte può assumere un valore decimale compreso tra 0 e 127.

# I canali MIDI

I canali MIDI possono essere paragonati ai canali televisivi: ad esempio se vedo in tv canale 5, ricevo i programmi che canale 5 diffonde via etere, e insieme a me li vedono tutti quelli che sono collegati e sintonizzati sullo stesso canale. Ovviamente chi è sintonizzato su RAI 1 non riceverà le informazioni che canale 5 trasmette e viceversa.

Allo stesso modo funziona per i canali MIDI, dove le informazioni passano attraverso il cavo di trasmissione. Ovviamente utilizzare un cavo MIDI per trasmettere un solo messaggio non sarebbe conveniente e per questo motivo una linea MIDI è stata divisa in 16 canali (logici) di comunicazione.

Ognuno di questi canali di comunicazione può trasmettere una determinata informazione che può essere ricevuta da un dispositivo sintonizzato su quel canale.

# I canali MIDI

La codifica digitale dei canali è quella più logica:

| Canale MIDI | Esadecimale | Binario |
|-------------|-------------|---------|
| 1           | 0           | 0000    |
| 2           | 1           | 0001    |
| 3           | 2           | 0010    |
| 4           | 3           | 0011    |
| 5           | 4           | 0100    |
| 6           | 5           | 0101    |
| 7           | 6           | 0110    |
| 8           | 7           | 0111    |
| 9           | 8           | 1000    |
| 10          | 9           | 1001    |
| 11          | A           | 1010    |
| 12          | B           | 1011    |
| 13          | C           | 1100    |
| 14          | D           | 1101    |
| 15          | E           | 1110    |
| 16          | F           | 1111    |

# Esempio di messaggio MIDI

Quando si preme un tasto in una master keyboard collegata con un cavo MIDI ad un expander, essa manda l'informazione al modulo sonoro relativa alla nota che è stata premuta, con quale intensità ed in quale ottava. Il messaggio in questione si chiama NOTE ON.

Lo status byte informa che è stata suonata una nota. Il numero della nota che deve essere suonata viene trasmesso dal primo data byte.

Se la tastiera è dinamica (cioè in grado di stabilire con quale intensità viene suonata una nota) allora quando si preme un tasto, oltre ai due messaggi precedenti se ne aggiunge un terzo di tipo data byte che contiene tale valore di dinamica nel range 0-127, altrimenti se la tastiera non è sensibile alla dinamica, viene inviato un valore di default (generalmente 64).



# Struttura dei messaggi MIDI

I messaggi MIDI si dividono in due categorie principali:

- Channel Messages;
- System Messages.

# Channel Messages

I messaggi di canale sono quelli che possono essere indirizzati ad uno qualsiasi dei sedici canali MIDI, essi si dividono in:

- Channel voice message;
- Channel mode message.

Sono Channel Voices Messages i messaggi di:

- Note on;
- Note off;
- Aftertouch (sia polifonico che di canale);
- Pitch bend;
- Program change;
- Control change (da 0 a 119).

# Note On e Note Off

Producono l'inizio e la fine di un evento sonoro. Questi messaggi contengono diverse informazioni fondamentali, sia per la loro logica di funzionamento che per la caratterizzazione del risultato sonoro. Un messaggio "Note On" contiene le seguenti informazioni:

- Numero di nota espressa in valori da 0 a 127.
- Velocità di pressione del tasto (dinamica di esecuzione, velocity-on), sempre con un'escursione numerica da 0 a 127.
- Informazioni di canale, da 1 a 16.

Nel caso di "Note Off" al posto del valore di velocity-on viene rilevata la velocità di rilascio (velocity-off). In realtà questo valore, nella maggior parte dei casi, viene ignorato.

# Note On e Note Off

La codifica utilizzata è la seguente:

| DEC | HEX | NOTE OFF - Evento nota spenta |
|-----|-----|-------------------------------|
| 128 | 80  | canale MIDI 1                 |
| 129 | 81  | canale MIDI 2                 |
| 130 | 82  | canale MIDI 3                 |
| ... | ... | ...                           |
| 143 | 8F  | canale MIDI 16                |

| DEC | HEX | NOTE ON - Evento nota accesa |
|-----|-----|------------------------------|
| 144 | 90  | canale MIDI 1                |
| 145 | 91  | canale MIDI 2                |
| 146 | 92  | canale MIDI 3                |
| ... | ... | ...                          |
| 159 | 9F  | canale MIDI 16               |

# After Touch

Come si intuisce dal nome, che tradotto significa "*dopo il tocco*", questo messaggio esprime i valori di pressione che si esercitano su di un tasto dopo che questo è stato premuto. I valori vanno, come sempre, da 0 a 127.

Ma che effetto produce l'Aftertouch?

In questo caso è difficile dare una risposta precisa, è possibile infatti assegnare a questo evento svariati "effetti". Tutto dipende dalla versatilità dello strumento: possiamo ad esempio fare in modo che ai valori generati dalla pressione sul tasto corrisponda un incremento del vibrato, oppure un aumento del volume, o ancora intervenire sui filtri del canale di sintesi.

# After Touch

La codifica utilizzata è la seguente:

| <b>DEC</b> | <b>HEX</b> | <b>AFTER TOUCH - Variazione della pressione di un tasto</b> |
|------------|------------|---|
| 160        | A0         | canale MIDI 1   |
| 161        | A1         | canale MIDI 2   |
| 162        | A2         | canale MIDI 3   |
| ...        | ...        | ...   |
| 175        | AF         | canale MIDI 16  |

# Pitch Bend

È stato e rimane uno dei controlli più utilizzati dai tastieristi nell'intento di simulare le capacità espressive di altri strumenti come la chitarra ed i fiati. Il Pitch Bend provoca come effetto sonoro la modifica dell'intonazione della nota suonata in maniera tanto più rilevante, quanto più ampia è l'azione esercitata sulla ruota o leva adibita al controllo.

La trasmissione dei dati MIDI relativi al Pitch Bend è un po' più complessa che in altri controlli, a causa della notevole quantità di dati che è necessario generare per assicurare la continuità e la fluidità che caratterizzano questo effetto.

Per questo motivo un messaggio di Pitch Bend è composto anziché dai due byte, caratteristici di quasi tutti i controlli, da tre byte. Il primo byte (di stato) è sempre riservato al rilevamento del tipo di controllo e del numero di canale MIDI. Il secondo ed il terzo byte (di dati) esprimono i valori d'intervento. La combinazione tra i due byte consente un'escursione totale di 16.384 step. Vista la necessità di dover creare Pitch Bend sia in senso ascendente che discendente, i valori definitivi saranno:

-8192 0 +8191.

# Pitch Bend

La codifica utilizzata è la seguente:

| <b>DEC</b> | <b>HEX</b> | <b>PITCH BENDER - Variazione della posizione del pitch bender</b> |
|------------|------------|---|
| 224        | E0         | canale MIDI 1   |
| 225        | E1         | canale MIDI 2   |
| 225        | E2         | canale MIDI 3   |
| ...        | ...        | ...   |
| 239        | EF         | canale MIDI 16  |



# Program Change

Viene utilizzato per inviare un "cambio di programma"; i valori possibili vanno da 0 a 127.

Visto il proliferare di tastiere, expander, effetti con a bordo spesso più di mille suoni, è facile immaginare come questo comando risulti del tutto insufficiente se usato da solo.

In genere l'abbinamento tra i Program Change Messages ed altri *Control Change* rende possibile superare il limite dei 128 suoni raggiungibili da questo controllo.

# System Messages

Questi messaggi, a differenza dei Channel messages, non contengono informazioni di canale, bensì di sistema, quindi possono essere ricevuti da qualsiasi apparecchiatura MIDI. Essi si dividono ulteriormente in:

- System common message;
- System Real time message;
- System exclusive message.

# La sintassi

Come primo esempio di messaggio MIDI, supponiamo di **suonare una nota**:

| Byte di stato                       |   | Byte di dati 1                           |   | Byte di dati 2                                  |
|-------------------------------------|---|--|---|---|
| NOTE ON<br>Nota accesa<br>144 (90H) | → | KEY NUMBER<br>Numero di nota<br>60 (3CH) | → | KEY VELOCITY<br>Pressione del tasto<br>64 (40H) |

Descriviamo questo primo esempio formato da 3 byte: con il primo si determina che è stato premuto un tasto, mentre con gli altri due il numero di nota e la velocità. Analizzando i numeri dei byte, vediamo che il 144 (90H, cioè 90 in esadecimale), corrisponde per il protocollo MIDI all'evento nota accesa e si riferisce specificatamente al canale MIDI n. 1, il secondo, numero 60 (3CH) al Do centrale, ed il terzo 64 (40H) alla pressione che corrisponde al volume con il quale suonerà la nota.

# La sintassi

Dopo il precedente messaggio la tastiera, l'expander, o altro che sia, suonerà la nota Do centrale fino a che non si invierà il seguente messaggio, **smettiamo di suonare una nota**:

| Byte di stato                        |   | Byte di dati 1                           |   | Byte di dati 2                                    |
|--------------------------------------|---|--|---|---|
| NOTE OFF<br>Nota spenta<br>128 (80H) | → | KEY NUMBER<br>Numero di nota<br>60 (3CH) | → | ATTACK VELOCITY<br>Rilascio del tasto<br>64 (40H) |

Supponendo che dopo aver premuto un tasto (note on), invece di rilasciarlo (note off) esercitassimo una pressione maggiore, seguirebbero messaggi del tipo:

| Byte di stato                                   |   | Byte di dati 1   |
|---|---|--|
| AFTER TOUCH<br>Pressione del tasto<br>208 (D0H) | → | AFTER TOUCH NUMBER<br>Numero di variazione<br>64 (40H) |

# La sintassi

Ma come mai il protocollo MIDI prevede solo 16 canali?

Come detto i messaggi MIDI viaggiano serialmente, cioè uno dopo l'altro a pacchetti di 8 bit. Sappiamo che i primi 8 bit inviati riguardano lo status Byte (byte di stato).

Per i messaggi di canale, fu deciso di assegnare il primo gruppo di 4 bit all'istruzione principale del messaggio ed il secondo gruppo di 4 bit alla codifica dei canali. Usando soltanto 4 bit, il massimo dei numeri che si possono rappresentare è, appunto, 16.

Per completezza, proviamo a mettere insieme un byte di stato, ad esempio **NOTA ON**:

| Decimale | Esadecimale | Binario  |
|----------|-------------|----------|
| 144      | 90          | 10010000 |

Come si è appena detto, il primo gruppo di 4 bit, in questo caso (1001), riguarda l'istruzione **NOTE ON**, ed il secondo gruppo di 4 BIT (0000), il numero di canale. Allo stesso modo vanno letti tutti gli altri byte di stato.

# I MIDI Files

Con la diffusione del MIDI e la nascita di nuovi software, si pone il problema della comunicazione dei dati precedentemente gestiti dagli strumenti musicali elettronici. Nasce allora il nuovo standard che uniforma il salvataggio dei dati per permettere questa comunicazione tra computer e strumenti, e tra software diversi: il MIDI file.

Prima della definizione dello SMF (Standard Midi File), proposto per la prima volta nel 1986 dalla americana Opcode Systems e adottato nel luglio 1988, i produttori di software musicale, per la memorizzazione di eventi MIDI, adottavano un loro particolare formato, con il risultato che, pur essendo i dati memorizzati, della stessa natura e su stessi supporti, i brani MIDI registrati dovevano essere letti dallo stesso tipo di software con cui erano stati creati.

L'implementazione e l'adozione dello SMF ha fatto in modo che, pur utilizzando software musicali diversi, la memorizzazione dei brani in formato MIDI avvenisse con la stessa procedura, consentendo così lo scambio e l'utilizzazione tra computer e sequencer diversi.

# I MIDI Files

Esistono tre diversi formati SMF:

## **Formato 0**

Tutte le tracce di un brano vengono mixate in una singola traccia che contiene però tutte le informazioni degli eventi relativi a tutte le tracce del brano.

## **Formato 1**

Le tracce vengono memorizzate in modo singolo e contengono gli stessi valori di tempo e metrica. La velocità del brano viene inserita nella prima traccia che fa da riferimento a tutte le altre.

## **Formato 2**

Le tracce vengono gestite indipendenti l'una dall'altra, con valori anche diversi di tempo e metrica.

Generalmente i più utilizzati sono il **Formato 0**, usato principalmente dai sequencer a lettura diretta, cioè che non devono caricare in memoria l'intera sequenza, ma prelevano ed eseguono i dati MIDI direttamente ed il **Formato 1** per i sequencer che possono creare e/o modificare SMF (specie i software musicali i quali hanno in genere gestiscono entrambi i formati).

# Il General MIDI GM

Il GENERAL MIDI (GM) nasce dunque per uniformare la risposta della varie apparecchiature agli stessi dati MIDI.

In altre parole, uno strumento che riceve un dato MIDI deve rispondere allo stesso modo di un altro, anche di costruttore differente.

Si noti, comunque, che usando apparecchiature MIDI differenti i suoni saranno simili, ma non identici, a causa soprattutto della diversa tecnologia usata dal costruttore.



# Il General MIDI GM

Di seguito, una parte della mappatura dei 128 suoni utilizzabili nel **GM**:

| <b>Numero Program Change</b> | <b>Strumento (Suono)</b> |
|------------------------------|--------------------------|
| 1                            | Acoustic Grand           |
| 2                            | Bright Acoustic          |
| 3                            | Electric Grand           |
| 4                            | Honky-Tonk               |
| 5                            | Electric Piano 1         |
| 6                            | Electric Piano 2         |
| 7                            | Harpsichord              |
| 8                            | Clav                     |
| ...                          | ...                      |

# m-LAN

**m-LAN**, e' un protocollo definito da Yamaha per lo scambio ad alta velocità di dati audio e MIDI. L'interfaccia utilizzata per garantire affidabilità ed elevate prestazioni è lo standard IEEE 1394, più conosciuto come FireWire, ideato da Apple Computer alla fine degli'anni '80.

L'elevata velocità di trasferimento dei dati, ha fatto in modo che in breve tempo questa interfaccia si imponesse nell'ambito della produzione video. Apple ha inoltre dotato tutti i suoi prodotti più recenti di porte FireWire, dando il via alla produzione, da parte di terzi, di periferiche, masterizzatori, Hard Disk, etc, compatibili con questo standard.

Yamaha, da parte sua, ha intravisto nella velocità e nella sperimentata affidabilità di FireWire, un sistema ideale per tutte quelle applicazioni che, in campo musicale, richiedono il trasferimento di grosse quantità di dati ed ha iniziato, già quattro anni fa, lo studio di un protocollo che permettesse di far viaggiare sui cavi FireWire i dati audio e MIDI. Questo protocollo ha preso il nome il nome di **m-LAN** (Music-LAN).

# m-LAN

Per dare un'idea delle reali capacità di **m-LAN** ecco alcuni dati:

Tramite interfaccia IEEE 1394 e' possibile raggiungere velocità di trasferimento pari a **400 Mbps** (**USB 1.1** raggiunge i 12 Mbps). Sono previsti aggiornamenti che consentiranno il transfert dei dati fino ad una velocità di **3.2 Gbps**.

La velocità attualmente supportata da **m\_Lan** e' di **200 Mbps**.

Con **m-LAN** si possono gestire, con un solo cavo, **16 canali X 256 porte**, per un totale di **4096 canali MIDI** e fino **100 canali Audio** (Mono).

Le periferiche possono essere collegate e scollegate senza che sia necessario spegnere e riavviare il computer. Si possono collegare in cascata fino a 63 devices. La rete rimane attiva anche a computer spento.

L'alta velocità del sistema, garantisce la massima precisione nelle operazioni di sincrono con registratori ed altre apparecchiature.

# m-LAN

Tutti possono avvalersi della tecnologia racchiusa in m-LAN senza sconvolgere il proprio parco macchine. La compatibilità con gli attuali standard è implementata ai massimi livelli. Oltre al pieno supporto del Midi, m-LAN è compatibile con il formato AES/EBU oltre che con quelle macchine che richiedono il sincronismo con Word Clock.

Attualmente, la produzione di apparecchiature in grado di supportare m-LAN è in mano a Yamaha e Korg (sua affiliata). Un esempio è il Korg Triton-rack in grado di supportare questo formato, tramite l'aggiunta della scheda opzionale EXB-mLAN, e che permette di importare ed esportare file audio a 48 Khz tramite due IN e 6 OUT, alla velocità di 200Mbps, oltre che gestire 16 canali MIDI.

Yamaha produce diversi sintetizzatori predisposti per l'aggiornamento a m-LAN, come l' S80, il C6X, e la serie EX, ed alcune schede sia per computer che per i suoi Mixer digitali 02R, 03D e 01v.