

{Musical Instrument Digital Interface MIDI}

APPUNTI PER IL CORSO DI INFORMATICA MUSICALE

INDICE DEI CONTENUTI:

Interfaccia, protocollo e sintassi MIDI	- 2 -
L'interfaccia	- 2 -
Protocollo e sintassi	- 3 -
Master, slave e soft thru	- 3 -
Messaggi MIDI	- 4 -
Canali MIDI.....	- 4 -
Struttura dei messaggi MIDI: channel and system messages	- 5 -
MIDI Files	- 7 -

Interfaccia, protocollo e sintassi MIDI

Il protocollo MIDI nasce negli anni 80 come frutto delle ricerche di D. Smith e C. Wood, gli stessi che nel 1981 pubblicheranno l'articolo titolato *The complete SCI MIDI*.

...MIDI, acronimo di **M**usic **I**nstrument **D**igital **I**nterface, ovvero, *interfaccia per il trasferimento digitale di dati tra strumenti musicali elettronici*. L'insieme, nello specifico, di una *interfaccia* fisica e, di uno *standard di comunicazione*, regole e messaggi interpretabili univocamente che permettono agli strumenti di scambiarsi informazioni.

Diventerà vero e proprio standard di comunicazione nel 1982 (con il MIDI 1 .0).

Nel 1983, il primo synth con interfaccia MIDI, il *PROFET 600* della SCI (Sequential Circuits).

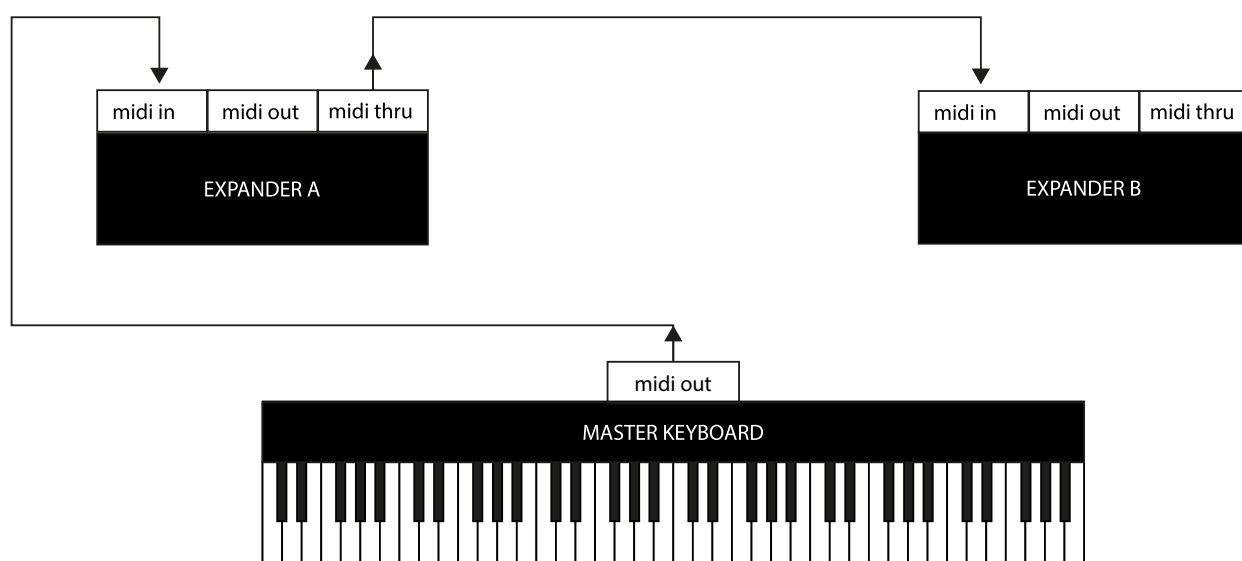
L'interfaccia

Nient'altro che un'interfaccia fisica che permette la trasmissione *seriale* di bit alla velocità di 31250 bit/sec in modo *asincrono*... asincrona, perché ogni sequenza di bit (messaggio MIDI) è delimitata da 2 bit (0, 1), il primo posto all'inizio della sequenza e, il secondo alla fine (start, end).

Una interfaccia MIDI è composta da una o più delle seguenti porte:

- MIDI In: riceve i dati in ingresso;
- MIDI Out: trasmette i dati in uscita;
- MIDI Thru: trasmette in uscita l'esatta copia dei dati ricevuti in ingresso. Serve, semplicemente, a collegare più dispositivi allo stesso dispositivo che invia dati.

Una possibile configurazione:



Protocollo e sintassi

I messaggi MIDI vengono trasmessi in gruppi di 8 bit a cui devono aggiungersi, come già anticipato, il bit di start e quello di stop; un'informazione trasmessa o ricevuta è dunque composta da 10 bit.

Vi sono due tipologie di byte MIDI:

- *status byte*: inviati per primi e servono a decifrare i byte di dati. Il bit più significativo dello status byte è uguale a 1... ne consegue che uno status byte può assumere un valore compreso tra 128 e 255. Trasmettono, in altre parole, il tipo di informazione.

1xxx xxxx **STATUS BYTE**

- *data byte*: byte di dati. Servono a comunicare i parametri necessari per un corretto funzionamento dello status byte. Il bit più significativo è pari a 0... ne consegue che un data byte può assumere un valore in base 10 compreso tra 0 e 127.

0xxx xxxx **DATA BYTE**

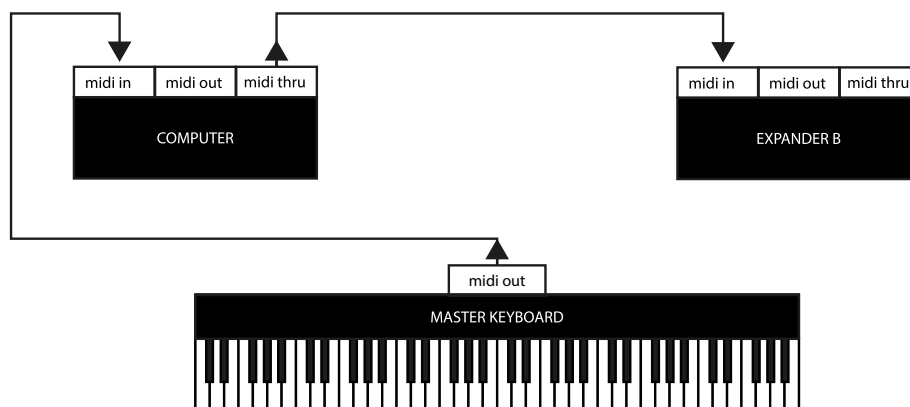
Master, slave e soft thru

Definiamo:

- *master*: tutti gli strumenti che pilotano altri strumenti;
- *slave*: tutti gli strumenti pilotati da altri strumenti.

...semplicemente, gli slave eseguono ciò che i master comandano.

- *soft thru*: un computer può comportarsi sia da master che da slave. Per evitare di dover ricablare ogni volta le macchine, è stata inserita la porta *soft thru*. Cerchiamo di capire meglio... ipotizziamo una configurazione del tipo:



il soft thru si comporta come TRHU quando a suonare è la tastiera e voglio ascoltare tramite expander; come OUT, invece, quando è il computer a comportarsi da master.

Messaggi MIDI

Sappiamo che con 8 bit possiamo codificare 256 valori diversi, i primi 128 per i data byte e, i secondi 128 per gli status byte. Ha la seguente struttura:

STATUS | DATA 1 | DATA 2

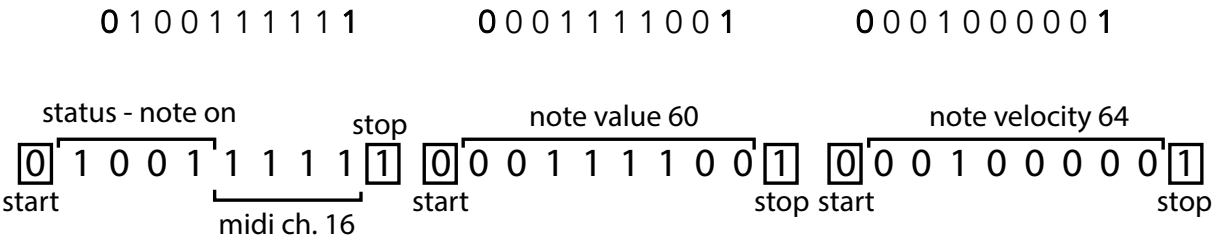
- Status = comando note on
- Data 1 = numero della nota
- Data 2 = valore velocity

Uno status byte ha questa forma:

1xxx nnnn

I primi 4 bit sono per l'istruzione principale; il secondo, per il numero di canale a cui inviare l'informazione contenuta nei primi quattro bit.

Un esempio di messaggio MIDI potrebbe essere il seguente:



Canali MIDI

Una linea MIDI possiede 16 canali logici di comunicazione, su ognuno dei quali può inviare una specifica informazione che riceverà il dispositivo sintonizzato su quello specifico canale. La codifica è la seguente:

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

Struttura dei messaggi MIDI: channel and system messages

I messaggi MIDI si dividono in due grandi categorie, *channel* e *system messages*:

- *channel messages*: possono essere inviati ad uno dei 16 canali disponibili e riguardano l'esecuzione musicale. Sono individuati dalle prime sette combinazioni dei bit xxx. Si dividono in:
 - *channel voice message*: sono generati durante l'esecuzione del brano e, sono:
 - **note on e note off**: producono l'inizio e la fine di un evento sonoro. Nello specifico caso del messaggio di *note on*, premendo un tasto della tastiera master, viene inviato un messaggio che presenta due data byte: il primo relativo all'altezza, il secondo alla velocity. In altre parole, si ordina ad uno slave di suonare una certa nota su di un certo canale. È necessario, pertanto che il messaggio specifichi:
 - il numero di nota espresso, da 0 a 127;
 - velocità di pressione del tasto (dinamica, *velocity-on*, la velocità con cui si preme o rilascia un tasto, solitamente usata per controllare il volume di una nota), con una escursione dinamica da 0 a 127;
 - il canale, da 0 a 16.

Un messaggio di *note off*, invece, ordina di smettere di suonare una certa nota su di un certo canale, non interrompe il suono, ma solo il suo invio. È un messaggio che si genera quando viene effettivamente rilasciato il tasto premuto in precedenza. Anch'esso è costituito da due data byte: il primo specifica la nota rilasciata, il secondo l'intensità di rilascio.

- **aftertouch polifonico:** esprime i valori di pressione che si esercitano su di un certo tasto dopo che questo è stato premuto, con una escursione che va da 0 a 127. Controlla vari parametri, come ad esempio, il vibrato, il tremolo, etc... L'aftertouch polifonico è simile al precedente, ma si riferisce al più complesso e costoso sistema di sensori indipendenti per ciascuna nota.
- **pitchbend:** modifica l'intonazione di una nota (glissando) in esecuzione, simula l'effetto *bending*. È un messaggio composto da 3 byte, il primo di stato, il secondo e il terzo dedicati ai valori di intervento. Può essere inviato via joystick o rotella; il range frequenziale dipende dallo slave, solitamente ± 2 semitoni.
- **program change:** serve ad inviare un cambio di programma (da 0 a 127 preset). Richiama il contenuto di una locazione di memoria.
- **control change (da 0 a 119):** serve ad inviare un cambio di controllo. Contiene 128 diverse variazioni di messaggio che fanno riferimento alle informazioni di espressione.
- **channel mode message:** riguardano il controllo generale di un dispositivo e non dei singoli canali. Controllano il modo di funzionamento delle voci di uno strumento. Sono:
 - **all sound off:** silenzia tutte le note che erano state attivate con il messaggio *note on*;
 - **all note off:** silenzia tutte le note... ferma tutto!
 - **local control:** attiva o disattiva la tastiera localmente in modo da permettere il controllo dei parametri del synth via MIDI IN da uno strumento esterno;
 - **omni mode off:** il dispositivo risponderà solo ai messaggi voice su un numero limitato di canali;
 - **omni mode on:** risponderà comunque al messaggio;
 - **mono mode on:** comanda di selezionare il modo mono;
 - **poly mode on:** comanda di rispondere polifonicamente ai messaggi in entrata.
- **system messages:** non contengono informazione di canale, ma di sistema e possono essere ricevuti da qualsiasi dispositivo MIDI. Sono individuati dall'ultima combinazione di bit 1111. Si dividono in:
 - **system common message:** danno istruzioni generali relative a tutto il sistema. Sono:
 - **midi timecode quarter frame:** messaggi nel formato ore, minuti, secondi e frame;

- **song position pointer:** indica il punto di una sequenza MIDI in cui deve posizionarsi il puntatore durante l'esecuzione;
- **song select:** selezione una song tra quelle disponibili nel dispositivo;
- **tune request:** intona due dispositivi.
- **system real time message:** organizzano la sincronizzazione del sistema. Essi sono:
 - **midi clock:** inviati 24 volte per quarto, restituiscono un tempo relativo, tempo musicale e, non assoluto come accade per il midi timecode quarter frame in formato SMPTE;
 - **start-continue-stop:** controllano i puntatori di tutti i dispositivi MIDI collegati;
 - **active sensing:** inviato ogni 300ms, serve a tenere attiva la connessione tra master e slave;
 - **system reset:** riporta tutti i dispositivi collegati ai valori predefiniti.
- **system exclusive message:** sono destinati alle informazioni specifiche di ogni produttore.

MIDI Files

SMF (*Standard Midi File*) è il formato standard universalmente accettato per il salvataggio e la trasmissione di informazioni MIDI... contiene tutte le informazioni e le note del brano in esso scritto.

Esistono 3 formati .smf (o .mid):

- **Formato 0:** tutte le tracce di un brano sono mixate in una singola traccia;
- **Formato 1:** le tracce sono salvate singolarmente e contengono gli stessi valori di metrica e tempo. Le informazioni di velocity si trovano nella prima traccia che fa da riferimento;
- **Formato 2:** tutte le tracce sono gestite indipendentemente una dall'altra, anche con valori di tempo e metrica diversi.