



grammen am obersten Qualitäts- und Preissegment gehören. Ich kann aber auch uneingeschränkt Produkte wie CakeWalk oder Logic ([abb.1](#)) empfehlen. Außerdem gibt es eine Menge kostenloser und ganz guter Programme im Internet zu finden. ([► Linktipps am Ende des Artikels](#))

Wie funktioniert so eine Software?

MIDI-Software zeichnet in erster Linie Informationen über die Art und Beschaffenheit eines zu reproduzierenden Klangereignisses auf. Eine MIDI-Note enthält somit eine Menge Informationen für ein Zielgerät oder eine Zielsoftware, wie zum Beispiel wann welcher Ton in welcher Tonhöhe und wie laut erklingen soll. Um MIDI-Noten in einer MIDI-Sequencer-Software zu erzeugen gibt es in der Regel zwei Möglichkeiten: Entweder macht man es "zu Fuß", malt also einfach die Noten als kleine Balken oder Punkte mit der Maus in ein Fenster der Software oder man spielt die Noten in Echtzeit über ein angeschlossenes MIDI-Keyboard ein. Beide Möglichkeiten sind legitim und unterliegen dem persönlichen Geschmack des jeweiligen Benutzers.

Wenn ich einen komplexen Streichersatz für ein Orchester einspielen muss, nehme ich lieber mein Keyboard. Aber um nun die eingespielten Notenwerte im Nachhinein zu editieren, benutze ich lieber die Maus meines Computers, da man über die grafische Benutzeroberfläche meistens sehr einfach und schnell alle MIDI-Parameter verändern kann. Obwohl die einzelnen Softwareproduzenten unterschiedliche Features und Oberflächen anbieten, ist die Herangehensweise im Prinzip immer dieselbe: Erst aufzeichnen, dann bearbeiten "bis es passt".

Sprechen Sie MIDI?

In meiner Praxis als Sounddesigner und Komponist für Computerspiele und Filme arbeite ich sehr viel mit MIDI-Technik. Es kommt immer wieder vor, dass einige Leute diese Technik nicht verstehen oder total verwechseln. Dieser Artikel soll Unklarheiten beseitigen und die Grundlagen von MIDI erläutern.

Oft habe ich mit Menschen zu tun, die MIDI-Daten mit Audio-Dateien verwechseln. MIDI ist aber nicht Audio (wie z.B. Wav-Dateien es sind), sondern ein genormtes Format um Soundmodule, Keyboards und einfach alles was Klang erzeugen, bearbeiten oder aufzeichnen kann zu steuern. Das heißt also, eine MIDI-Note teilt dem jeweiligen Gerät mit, was genau es tun soll. Diese Steuerdaten lassen zum Bei-

spiel einen Klangerzeuger Töne produzieren oder ein Effektgerät verschiedene Parameter verändern. MIDI als solches kann aber selber keinen Klang erzeugen.

Um MIDI-Steuerdaten zu erzeugen bzw. aufzuzeichnen, gibt es eine Menge an Software, so genannte MIDI-Sequencer. Ich selber arbeite am liebsten mit Cubase und ProTools, welche beide zugegebenermaßen zu den Pro-



abb.1 Screenshot des MIDI- und Audiosequenzers Logic 6.
(www.emagic.de)

Die wichtigsten MIDI-Parameter im Überblick:

Note On und Note Off

Wenn wir eine Taste am Keyboard anschlagen wird von unserer MIDI-Sequencer-Software ein "Note On" Befehl registriert. Dieser ist so lange aktuell bis wir die Taste wieder loslassen. Dann wird nämlich ein "Note Off" Befehl aufgezeichnet. Ab und an kann es vorkommen, dass Soundmodule oder Keyboards einen Ton wiedergeben und dieser plötzlich "hängen bleibt", sprich ununterbrochen ertönt obwohl wir keine Taste mehr berühren. Wenn dies passiert, ist der "Note Off" Befehl auf seinem Weg zwischen Keyboard, MIDI-Software und Klangerzeuger irgendwo verloren gegangen, oder wurde zum Beispiel aus Versehen von uns aus der MIDI-Steuersoftware gelöscht.

Kanalnummer

Wenn wir eine Taste am Keyboard drücken, wird auch die Information darüber aufgezeichnet, welcher MIDI-Kanal angesprochen werden soll. Genauso sendet unsere MIDI-Sequencer-Software eine Information an den jeweiligen Klangerzeuger, für welchen MIDI-Kanal die folgenden No-

tenwerte bestimmt sind.

Der MIDI-Standard sieht vor, dass auf 16 Kanälen gesendet und empfangen werden kann. In der Praxis heißt das, dass wir 16 verschiedene Instrumente pro MIDI-Schnittstelle direkt ansprechen können. Je nach den Eigenschaften (Mono- oder Polyphonie) des anzusprechenden Klangerzeugers, können wir jeden Sound auch mehrstimmig spielen. Zum Beispiel 6-stimmige Klavierakkorde auf Kanal 1, einen einstimmigen E-Bass auf Kanal 2, usw. Wenn wir gleichzeitig mehr als 16 Kanäle (=Instrumente) benötigen, müssen wir in unseren Rechner, ein weiteres MIDI-Interface einbauen. Es gibt bereits MIDI-Einbaukarten für Rechner, die mehrere Ein/Ausgänge haben und somit mehrere 16-Kanal Instrumentengruppen ansprechen können. Wenn wir allerdings nur eine MIDI-Schnittstelle zur Verfügung haben, gibt es auch die Möglichkeit zu "schummeln". Es ist möglich, auf einem MIDI-Kanal abwechselnd mehrere Instrumente anzusprechen. Dies funktioniert mit dem "Program-Change" Befehl.

Program-Change

Der Program-Change Befehl teilt dem Klangerzeuger mit, welcher Sound/Klang nun erklingen soll. Da es bei einigen Instrumenten oft der Fall ist, dass sie nur ab und an spielen, kann man den Program-Change Befehl dazu benutzen um auf einem Kanal mehrere verschiedene Klänge, bzw. Instrumente, anzusprechen.

Ein Beispiel: Ich habe einen Song der Bass, Schlagzeug, Rhythmusgitarre und 2 Soli, abwechselnd mit einem Piano und einem Orgelsound gespielt, beinhaltet. Ich lege den Bass auf Kanal 2, das Schlagzeug auf Kanal 10, die Rhythmusgitarre auf Kanal 3. Kanal 1 belege ich mit einem Klaviersound für das Solo. Wenn nun das Klaviersolo zu Ende ist, baue ich einen "Change-Programm-to-Orgelsound" Befehl ein. Danach kann ich über Kanal 1 direkt den Orgelsound antriggern, und mein Solo wird daraufhin von einem Orgelsound wiedergegeben.

Diese Vorgehensweise spart eine Menge Spuren, wenn man Instrumente hat, die nicht alle zur selben Zeit erklingen müssen.

Der "Program-Change" Befehl ist natürlich nicht nur dazu da, um zwischendurch mal den Sound zu wechseln oder auch Effekteinstellungen anzusprechen. Er ist vor allem auch dazu da, um von Anfang an festzulegen auf welchem Kanal nun welcher Klang angesprochen werden soll.

MIDI-Lautstärke

Der MIDI-Lautstärke-Parameter teilt dem jeweiligen Klangerzeuger mit, wie laut ein Ton erklingen soll, wie stark also eine Taste angeschlagen wurde. Dies kann in 128 Einzellautstärkeschritten erfolgen. Dabei steht der Wert 0 für keinen Klang, der Wert 127 für die größtmögliche Lautstärke.

Control-Change

Mittels des Control-Change Befehles werden verschiedene Parameter verändert/angesprochen wie zum Beispiel eine Veränderung des Modulationsrades am Keyboard, oder der Einsatz eines Fußpedals.

Sys-Ex Datendump

"Sys-Ex" steht für "System Exclusive Message". Diese Art von Daten nehmen im MIDI-Protokoll eine Art Sonderstellung ein, denn abgesehen vom Format der übertragenen Datenbytes gibt es für Sie keinerlei Norm. Jeder Hersteller von Klangerzeugern hat nun die Möglichkeit, diese Art der Daten für sich selber zu verwenden, um eigene Befehle zu definieren. Somit können sich Soundmodule von ROLAND miteinander "unterhalten", während eine Soundkarte von Yamaha diesen "Slang" nicht verstehen würde.

In der Regel benutzt man aber diese Art von Daten zur Sicherung von Einstellungen an den Klangerzeugern und Effektgeräten. Wenn ein Gerät beschädigt wird, kann es schon mal vorkommen dass alle im Gerät gespeicherten Daten und Einstellungen verloren gehen. Mittels Sys-Ex Dumps gibt es die Möglichkeit Sicherheitskopien dieser Daten auf der Computerfestplatte oder auf eigenen Speicherkarten des Herstellers anzufertigen.

Die MIDI-Hardware

Die MIDI-Schnittstelle (auch "MIDI-Port" genannt) ist eine runde Buchse mit 5 Pins (abb.3). Es gibt diesen Port in 3 Ausführungen: MIDI-In, MIDI-Out und MIDI-Thru. Die Namen verraten schon die Ei-



abb.3
Die standardisierte MIDI-Buchse mit ihren fünf Pins.

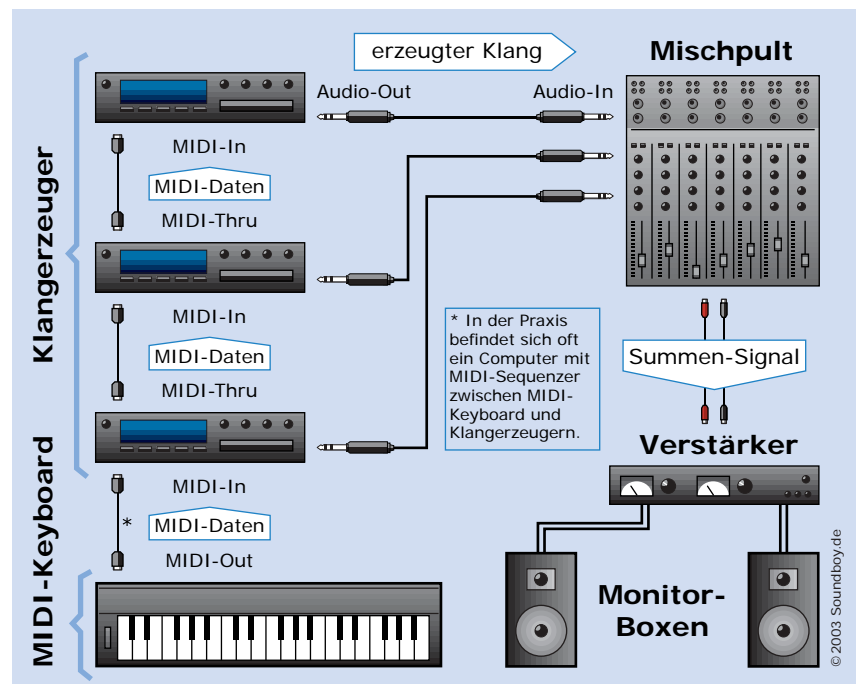


abb.2 Durch die MIDI-Thru Schnittstelle ist es möglich, mehrere Klangerzeuger über nur eine Leitung gleichzeitig anzusprechen.

genschaften dieser "Schnittstellen": Ausschließlich MIDI-Outs finden wir in erster Linie bei Geräten, welche nur Daten erzeugen wie z.B. Master-Keyboards, MIDI-Fußleisten oder MIDI-Schalter.

MIDI-Ins finden wir bei Geräten, die MIDI-Daten empfangen und verwerten, also z.B. bei Klangerzeugern oder Effektgeräten.

Am MIDI-Thru Port liegen immer dieselben Daten, die auf dem MIDI-In ankommen. MIDI-Thru ist also so eine Art "Verteilerstück". Somit ist man nicht nur auf ein einziges MIDI-Gerät beschränkt, das man über seine MIDI-Out Schnittstelle ansprechen kann. Wenn mehrere MIDI-Geräte angetriggert werden sollen, kann man diese über MIDI-Thru als Kette betreiben, und zum Beispiel mittels nur eines Masterkeyboards vier Soundmodule antriggern.

Die meisten neueren MIDI-fähigen Geräte (egal ob es Effektgeräte, MIDI-Steckkarten für den Computer, Keyboards oder Soundmodule sind) besitzen bereits alle 3

Arten von MIDI-Buchsen. Viele Geräte besitzen des Weiteren noch eine oder mehrere LEDs um "Incoming" oder "Outgoing" MIDI-Daten optisch anzuzeigen.

MIDI-Geräte werden nun über die MIDI-Ports mittels MIDI-Kabel miteinander verbunden. Diese Kabel gibt es in fast jeder Länge und Farbe im Musikfachhandel zu kaufen. Jedoch sollten die MIDI-Kabel 15 Meter Länge nicht überschreiten, da sonst einfach zu viele Übertragungsfehler auftreten würden.

Zahlen, Zahlen, Zahlen...

Wenn wir nun MIDI-Daten genauer betrachten, ist es unerlässlich sich mit etwas Mathematik zu beschäftigen. Aber keine Angst, es wird sicher nicht zu komplex. Es ist nicht unbedingt notwendig all die folgenden mathematischen Grundgesetze von MIDI zu beherrschen, jedoch fällt es auf jeden Fall leichter das Protokoll "MIDI" besser zu verstehen wenn man sich etwas mit der Materie vertraut macht. Im täglichen Leben

benutzen wir das dezimale Zahlensystem, auch Zehnersystem genannt, welches aus den zehn Zahlen 0 - 9 besteht. Es lassen sich als zehn verschiedene Werte darstellen. Bei Werten größer 9 werden diese Zahlen nun miteinander kombiniert.

Auf die gleiche Art und Weise setzen sich in der "digitalen Welt" Zahlen zusammen, nur mit dem Unterschied dass diese mit dem Binärsystem, auch Dualsystem genannt, dargestellt werden. Dieses stellt Zahlen nur mithilfe der Werte 0 und 1 dar.

Da digitale Schaltungen nur die Zustände "0" (kein Strom) und "1" (Strom an) kennen, erklärt sich somit die Verwendung des binären Zahlensystems. Dies erschwert aber nun die Darstellung von Zahlen, da ungleich mehr "Stellen" verwendet werden müssen als im dezimalen Zahlensystem um eine Zahl darzustellen und eine Zahl nun aus einer längeren Reihe Nullen und Einsen besteht.

Eine einzelne Stelle so einer "Nuller-Einser-Zahl" wird in der Computerfachsprache als "ein Bit" bezeichnet, eine Reihe von 8 Bits als "Byte". Nun zu erklären wie eine Dualzahl dargestellt wird; würde den Rahmen dieses Artikels sprengen. Mehr Informationen dazu sind im Internet auf

diversen Websites zu finden die das Thema umfassend behandeln.

Für uns ist nur wichtig zu wissen dass mittels eines Bytes, also 8 Bits, die Zahlen 0-255 darstellbar sind.

Was passiert im MIDI-Kabel?

Da MIDI-Signale nun digitaler Natur sind und seriell (nacheinander) in unserem MIDI-Kabel übertragen werden, und die "Übertragungsbreite" 31.250 Bits (auch "Baud" genannt) pro Sekunde beträgt, heißt das für uns, dass ein einzelnes Bit 32 Mikrosekunden benötigt um durch das MIDI-Kabel zu "wandern". Wenn keine Daten übertragen werden, also die MIDI-Strecke im "Ruhezustand" ist, liegt eine Spannung von 5 mA Strom im Kabel an. Dies wird als "Zustand 1" gewertet. (Wir erinnern uns: 0 = kein Strom, 1 = Strom).

Jedem nun gesendeten Byte wird ein "0"-Startbit vorangestellt. Somit weiß der MIDI-Empfänger, dass nun als nächstes MIDI-Daten folgen werden. Nach 1,5facher Zeiteinheit, also 48 Mikrosekunden, wird nach unserem "Achtung-jetzt-kommt-was!"-Bit das erste Datenbit übermittelt. Nach 2,5facher Zeiteinheit ist das zweite Bit an der Reihe, nach 3,5facher

Zeiteinheit das Dritte, und so weiter. Dies wird solange vollzogen, bis alle 8 Datenbits (unser erstes Byte) übertragen sind. Nun wird ein Stop-Bit (diesmal "1") gesendet. Somit wird die Leitung wieder in den 5 mA Ruhezustandstrom versetzt.

Da MIDI-Daten einfach aus einer Reihe von Zahlen bestehen, können somit alle MIDI-Parameter übertragen werden.

Das MIDI-Datenformat

Ein einzelnes Byte ist natürlich viel zu wenig um alle bei einer MIDI-Übertragung anfallenden Daten darzustellen. Daher bestehen so genannte "MIDI-Messages" aus 3 Bytes: Einem Statusbyte und 2 Datenbytes. Das zuerst gesendete Statusbyte enthält als Merkmal auf der ersten Stelle eine "1" als Startbit. Des weiteren enthält nun das Statusbyte Informationen über die Art der nun folgenden Datenbytes, wie zum Beispiel "dies sind nun Programm-Change-Daten", oder "dies sind nun Note-On-Befehle".

Die Datenbytes beginnen immer mit einer "Null" an erster Stelle. Hat nun der MIDI-Empfänger ein ankommendes Byte auf Grund der führenden "1" als Statusbyte identifiziert, wird dieses in 2 gleichlange Stücke von je 4 Bits, so genannte "Nibbles" geteilt.

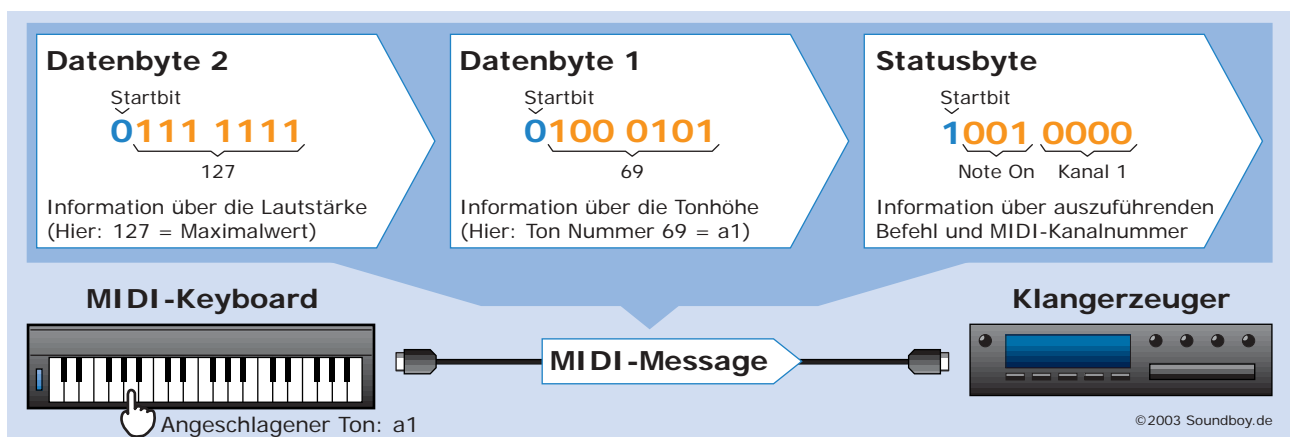


abb. 4 Diese Abbildung zeigt das Schema der Übertragung einer MIDI-Message. Nach dem Statusbyte folgen zwei Datenbytes, welche die eigentliche Information über das Klangereignis enthalten.

Das erste Nibble gibt nun über die Art des MIDI-Befehles Auskunft. Da Bit 1 (also unser "Startbit") bereits verwendet wurde, bleiben noch drei Bits übrig, welche folgendermaßen kodiert sein können:

000 = Note Off Befehl
001 = Note On Befehl
010 = Poly Key Pressure
101 = Channel Pressure
110 = Pitch Wheel Change
011 = Control Change
100 = Program Change
111 = System Realtime

Das zweite Nibble des Statusbytes teilt dem Empfänger die senderseitig eingestellte MIDI-Kanalnummer mit. Weil dafür 4 Bits zur Verfügung stehen, können 16 Kanäle definiert, also angesprochen werden. Somit haben wir auch die Lösung des Rätsels, warum pro MIDI-Schnittstelle nur 16 Instrumente/Kanäle verwendet werden können.

Ein Beispiel: Wenn wir auf unserem MIDI-Keyboard den Ton a1 anschlagen, wird folgende MIDI-Message über die MIDI-Out Schnittstelle zu unserem Klangerzeuger gesendet:

10010000 (Statusbyte)
01000101 (1. Datenbyte)
01111111 (2. Datenbyte)

Da eine angeschlagene Taste einen "Note On"-Befehl auslöst, steht im ersten Nibble unseres Statusbytes nach der dem Startbit ("1") die Zahlenfolge "001" für "Note-On". Der MIDI-Kanal 1 wird dann durch das zweite Nibble ("0000") dargestellt. Normalerweise ist die Dualzahl "0000" auch die Dezimalzahl "0", aber die Nummerierung der MIDI-Kanäle erfolgt nicht von "0-15", sondern von "1 - 16". Wir müssen also zu der sich aus dem zweiten Nibble ergebenden Dezimalzahl immer eine Eins zuzählen.

Die nun folgenden Datenbytes werden, im Gegensatz zum Statusbyte nicht zerteilt, sondern in voller Breite ausgewertet. Weil aber das erste Bit eines Datenbytes aus Datenerkennungsgründen (siehe weiter oben) immer auf "0" steht, bleiben für die Datenübermittlung lediglich 7 Bits übrig, sodass sich die Werte 0 - 127 darstellen lassen.

Der Wert des ersten Datenbytes "(0)1000101" ergibt dezimal umgerechnet die Zahl 69. Dies ist auch die Nummer unseres "a1"-Tones. Somit weiß der Empfänger welcher Ton nun erklingen soll.

Im zweiten Datenbyte "(0)1111111" wird definiert, wie laut der Ton erklingen soll. Dieser Wert ergab sich daraus, wie fest der Ton auf dem Keyboard angeschlagen wurde. "1111111" ergibt als dezimaler Wert 127, also die höchstmögliche Lautstärke.

Beim loslassen der Taste wird nun wieder ein 3-fach Byte gesendet. Diesmal mit einem "Note-Off"-Befehl und den Informationen für welche Note er gilt (in unserem Fall wieder a1) und wie schnell die Taste losgelassen wurde.

Autor: Alexander Loitsch
([▶ www.sound4you.net](http://www.sound4you.net))

Layout/Grafik:
Dennis Hölscher
([▶ www.hellfish-design.de](http://www.hellfish-design.de))

Stand: September 2003
Alle Rechte vorbehalten.

[▶ Weitere Artikel gibt's auf \[www.soundboy.de\]\(http://www.soundboy.de\)](#)

Linktipps

- 1 www.zem-college.de/midi/
Kompendium zum Thema MIDI
- 2 www.hitsquad.com
Riesiges Musik-Software Archiv
- 3 www.musicrobot.com
MIDI-File Suchmaschine