

Klang ist Allgegenwärtig. – Sprechen Sie „MIDI“? (Teil 2)

(Von Alexander „loitschix“ Loitsch, © 2003, <http://sound4you.net>)

1. MIDI im Detail, oder „Wie sieht denn das von Innen aus?“

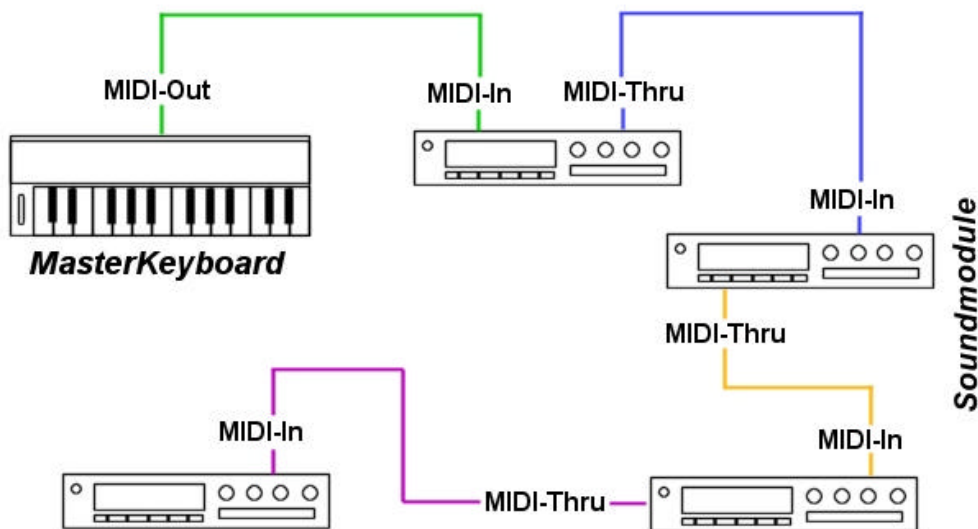
In Teil 1 dieses MIDI-Artikels habe ich eine kurze Einführung in die Materie und einen Überblick über die wichtigsten MIDI-Parameter verfasst. In diesem Artikel nun werden die theoretischen Hintergründe etwas genauer beleuchtet, sowie einiges über MIDI-Hardware geschrieben.

2. Die MIDI-Hardware oder „Was kann man womit zusammenstöpseln?“.



Die MIDI-Schnittstelle (auch „MIDI-Port“ genannt) ist eine runde Buchse mit 5 Pins (siehe Abbildung). Es gibt diesen Port nun in 3 Ausprägungen: MIDI-In, MIDI-Out und MIDI-Thru, und diese Namen verraten schon die wichtigsten Eigenschaften dieser „Stecker“:

Ausschließlich MIDI-Outs finden wir in erster Linie auf Geräten welche nur Daten erzeugen (Master-Keyboards, MIDI-Fußleisten oder MIDI-Schalter), d.h. hier werden ausschließlich MIDI-Daten gesendet. MIDI-In ist dazu da um MIDI-Daten zu empfangen, und 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 4 Soundmodule antriggern. (Siehe Abb. 2).



(c) 2003 by Alexander M. Loitsch

Abb. 2

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 übersteigen, da die Daten im Inneren auf Basis von unsymmetrischen Stromimpulsen als Stromschleife erfolgt und ab einer gewissen Länge einfach zu viele Störimpulse bzw. kapazitive Effekte auftreten könnten was, einfach gesagt, Übertragungsfehler verursachen würde.

3. 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 wird 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 das diese mit dem Binärsystem, auch Dualsystem genannt, dargestellt werden welches aus nur 2 Zahlenwerten, 0 und 1, besteht.

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 das mittels eines Bytes, also 8 Bits, die Zahlen 0-255 darstellbar sind.

3. 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 das ein einzelnes Bit 32 Mikrosekunden benötigt um durch die MIDI-Leitung 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 das 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, usw.... 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 Ruhezustandsstrom“ versetzt.

Da MIDI-Daten (siehe nächstes Kapitel) einfach aus einer Reihe von Zahlen bestehen können somit alle MIDI-Parameter übertragen werden.

4. Das MIDI-Datenformat

Ein einzelnes Byte ist natürlich viel zu wenig um alle bei einer MIDI-Übertragung anfallenden Daten darzustellen. Daher bestehen sogenannte „MIDI-Messages“ aus 3 Bytes und zwar einem Statusbyte, und 2 Datenbytes. Das zuerst gesendete Statusbyte enthält als Merkmal auf der linken, also ersten Stelle, eine „1“ (Siehe weiter oben: „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“, etc...

Die Datenbytes beginnen immer mit einer „Null“ als erste (linke) 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, sogenannte „Nibbles“ zerteilt. 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 nun dem Empfänger die senderseitig eingestellte MIDI-Kanal Nummer mit. Da dafür 4 Bits zur Verfügung stehen, können 16 Kanäle „dargestellt“, also „angesprochen“ werden.

Somit haben wir auch die Lösung des Rätsels von Teil 1 unseres MIDI-Artikels warum pro MIDI-Schnittstelle nur 16 Instrumente/Kanäle verwendet werden können.

4. Ein Beispiel:

Wenn wir nun auf unserem Masterkeyboard eine Ton anschlagen (zum Beispiel den Ton a1 mittels eines Klavierklanges auf Kanal 1) wird folgende MIDI-Message über die MIDI-Out Schnittstelle zu unserem Klangerzeuger gesendet:

1 0 0 1 0 0 0 0	0 1 0 0 0 1 0 1	0 1 1 1 1 1 1 1
Statusbyte	1. Datenbyte	2. Datenbyte

Da eine angeschlagene Taste einen „Note On“ Befehl auslöst steht in unserem ersten Nibble unseres Statusbyte nach der „1“ (dem Startbit) 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 (linke) 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 (1000101) ergibt dezimal umgerechnet die Zahl 69. Dies ist auch die Nummer unseres „a1“-Tones. Somit weiß der Empfänger nun welcher Ton erklingen soll. Im zweiten Datenbyte (1111111) wird nun definiert wie laut der Ton erklingen soll. Dieser Wert ergab sich daraus wie fest der Ton auf der Tastatur angeschlagen wurde. 1111111 ergibt als dezimaler Wert 127, also die höchste Lautstärke die möglich ist.

Beim loslassen der Taste wird nun wieder ein 3-fach Byte gesendet, diesmal mit der Info „Note-Off“, und den Informationen für welche Note er gilt (in unserem Fall a1) und wie schnell die Taste losgelassen wurde.

5. Genug MIDI...

Ich hoffe dass nun der Schleier über einigen Grundlagen des Themas MIDI etwas gelüftet wurde. Ich habe in absehbarer Zeit noch einen dritten Teil zum Thema MIDI geplant um noch tiefer in die Materie einzudringen.

Ich möchte mich auf diesem gerne nochmals bei Manfred Eitler aus Wien/Österreich für seine konstruktive Kritik und das aufmerksam Lesen meiner Beiträge bedanken! ☺

Bei Fragen oder Anregungen zu diesem Artikel kontaktieren Sie mich bitte unter
a.loitsch@sounddesign.loitschix.com.