

MIDIFormat



be-sign.net

bernhard r. scheurer
mainstrasse 2 • 6430 erzhausen
www.be-sign.net • info@be-sign.net



jms

jugendmusikschule erzhausen
industriestr. 6-8 • 6430 erzhausen
www.jms-erzhausen.de • info@jms-erzhausen.de

- Übersicht
- Basics
- Syntax
- Hardware
- Anwendungen
- Wörterbuch
- Zukunftsmusik
- Materialien
- DIY
- Anhang

**Ziel
dieses
Workshops**

Anfang der 80er war es nur mit erheblichem Aufwand möglich, Audiodaten digital aufzuzeichnen und zu verarbeiten. Die Lösung dieses Problems hieß MIDI, war technisch relativ anspruchslos aber enorm leistungsfähig, was das Verarbeiten von Steuerdaten angeht. Damit wurde der breiten Masse der Musiker völlig neue Möglichkeiten eröffnet.

Während Audiodaten einen ständigen Datenstrom erzeugen, fallen bei MIDI nur dann Daten an, wenn eine Taste, ein Regler, ein Pedal oder ähnliche bewegt wird. Die aufgezeichneten Daten sind unabhängig vom Klang und können deshalb universell verwendet werden. So viel zu den Vorteilen.

Gemäß dem damaligen Stand der Technik löst MIDI mit einer "Genauigkeit" von 127 Stufen auf. Das reduziert den Datenstrom erheblich, bringt aber auch ein paar musikalische Einschränkungen mit sich. Stufenlose Änderungen in z.B. Tonhöhe oder Lautstärke, lassen sich so nicht befriedigend abbilden.

Ungeachtet dessen, hat sich MIDI bis heute gehalten. Mit der aktuellen Technik lässt sich auch die eine oder andere Grenze aufheben, bzw. umgehen. Es ist wohl der einfachen und robusten Struktur geschuldet, dass MIDI bis heute eine feste Größe ist.

Ziel dieses Workshops ist es, Geschichte, Aufbau und Anwendung von MIDI kennen zu lernen und zu verstehen.

Not
macht
erfinderisch

In den 80er Jahren hat es Computer hat es bereits ins heimische Wohnzimmer geschafft, ist also bezahlbar geworden. Als Anwender gebietet man über eine CPU zwischen sagenhaften 1-4MHz Taktfrequenz und bis zu 64Kbyte(!) RAM. Parallel hierzu erobern die ersten bezahlbaren digitalen Synthesizer den Markt. Zu den bekanntesten Vertretern jener Generation gehören sicherlich Yamaha DX7 und später Korg M1.

MIDI 1.0 wurde 1981 entwickelt und im August 1982 eingeführt. Das robuste und flexible Protokoll hat zwischenzeitlich einige Erweiterungen erfahren und ist aus der Studio- und Livetechnik nicht mehr weg zu denken. Die beiden Entwickler Dave Smith (Sequential Circuits) und Ikutaro Kakehashi (Roland Corporation) wurden 31 Jahre später für ihre Leistung mit dem technischen Grammy ausgezeichnet. Das Protokoll wird bis heute von der MIDI Manufacturers Association (MMA) gepflegt und weiter entwickelt.



Dave Smith



Ikutaro Kakehashi

Gemessen an den damaligen Möglichkeiten ist MIDI (Musical Instrument Digital Interface) schlicht genial. Um es zu verstehen, ist es allerdings notwendig, sich mit der Zahlenwelt der Computer zu beschäftigen, also mit dem Binär- und dem Hexadezimal-System. Aber keine Panik, das ist einfacher, als die Begriffe vermuten lassen.

zehn,
zwei
und
sechzehn
Finger

Dezimal: Wenn wir zählen, verwenden wir 10 verschiedene Ziffern in festgelegter Reihenfolge: 0, 1, 2, 3, 4, 5, 6, 7, 8 und 9. Sind wir bei der letzten Ziffer angekommen, fügen wir vorne eine Stelle hinzu (die uns über die Anzahl der bereits erfolgten Durchläufe informiert) und beginnen dahinter wieder mit 0. So kommt nach der 9 die 10 und nach der 99 die 100. Dieses System ist praktisch, haben wir zur Veranschaulichung doch auch 10 Finger zur Verfügung.

Binär: Was wäre, wenn wir nur zwei Finger hätten? Nun, das 10-Finger-System läßt sich vortrefflich auf zwei Finger übertragen. Diese entsprechen dann den Ziffern 0 und 1. Wir zählen wie gewohnt durch. Sind wir bei der letzten Ziffer angekommen, fügen wir vorne eine Stelle hinzu und beginnen wieder von neuem: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, ...

Hexadezimal: In MIDI-Dokumentationen trifft man hin und wieder auch auf „hexadezimale“ Zahlen, gewissermassen ein 16-Finger-System mit den Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E und F. Es gelten die selben Regeln wie in anderen Zahlensystemen auch.

Fazit: Je mehr Ziffern wir verwenden, desto weniger Platz brauchen die dargestellten Zahlen: die Zahl 127 benötigt binär 7 Zeichen (1111111), dezimal drei (127) Zeichen und im hexadezimal nur noch 2 Zeichen (7F).

Dezimal	Binär	Hexadezimal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
...	...	...
127	1111111	7F
...	...	...
254	11111110	FE
255	11111111	FF

Licht
an
Licht
aus!

Es liegt in der Natur digitaler elektronischer Geräte, nur mit zwei Fingern zählen zu können. Die beiden benötigten Ziffern 0 und 1 werden durch "kein Strom" und "Strom" dargestellt. Diese kleinsten Informationen nennt man Bit (binary Digit). Praktischerweise fasst man 8 Bit zu einem Byte zusammen; so lassen sie sich leichter handhaben. Und so wohnen die Bits in ihren Bytes:

Bit-Nummer	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1
binärer Wert	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
dezimaler Wert	128	64	32	16	8	4	2	1

Um einen dezimalen Wert zu speichern, belegen wir einfach die benötigten Bits mit einer 1 und lassen die restlichen Bits leer (0). Um z. B. die Zahl "5" zu speichern, füllen wir Bit 1 und Bit 3 mit einer "1". Bit 1 hat den Wert "1" und Bit 3 hat den Wert "4" - zusammen also "5":

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

Um den dezimalen Wert 75 im Speicher abzulegen, benötigen wir die Bits 1, 2, 4 und 7. Denn: $1+2+8+64=75$

0	1	0	0	1	0	1	1
---	---	---	---	---	---	---	---

Mit einem Byte lässt sich maximal der dezimale Wert 255 speichern; in den 80er Jahren der Standard in der Computertechnik. Demgegenüber steht nun die ganze Fülle an musikalischen Informationen, die mit diesen 255 Werten zu verwalten sind. Klingt unmöglich? Ist es aber nicht. Was man mit diesen 8 Bit alles bewerkstelligen kann, folgt auf der nächsten Seiten ...

- Nachrichten
- Ein Beispiel
- SysEx-Magie

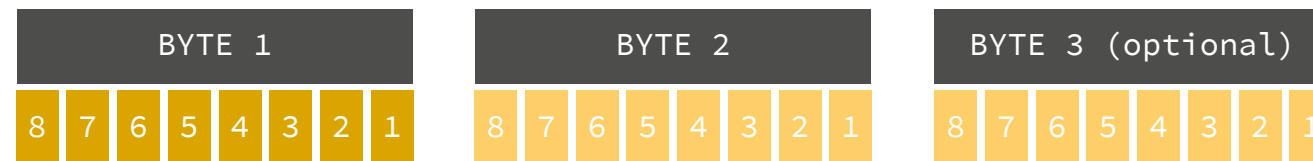
Bytes
im
Dienste
der
Kunst!

Zwei Vorgaben standen bei der Entwicklung von MIDI im Raum:

Zum einen wollte man musikalischen Informationen transportieren. Da die damalige Technik keine Möglichkeit bot, Audio-Material in sinnvoller Geschwindigkeit zu übertragen, geschweige denn in Echtzeit zu bearbeiten, musste ein Weg gefunden werden, die anfallenden Datenmengen in eine einfache Sprache zu übertragen.

Zum anderen stand eine Auswahl von lediglich 255 verschiedenen Werten zur Verfügung, um die zu übertragenden Informationen in eine logische Form zu bringen. Dazu gehören sowohl rein musikalische Informationen, als auch Speicherinhalte, Steuer- und Klangdaten der elektronischen Musikinstrumente.

MIDI besteht aus Nachrichten, die fast alle nach dem selben Prinzip aufgebaut sind: Ein **Statusbyte**, das die Art der Information beschreibt und ein bis zwei **Datenbytes**, die die zugehörigen Informationen beinhalten.*



Um **Status** und **Daten** auseinander halten zu können, wird Bit 8 als Markierung benutzt. Ist es belegt, handelt es sich um einen Status, ist es leer, handelt es sich um Daten. Da man mit den Bits 1-7 maximal die Zahl 127 abbilden kann, ist das auch der höchste Datenwert, der im MIDI-System vorkommt.

Manche Status-Bytes transportieren in den Bits 1-4 noch eine Kanal-Nummer. So ist es möglich, mit einem MIDI-Kabel auf 16 verschiedenen Kanälen zu kommunizieren.

* Jedes Byte wird von einem Start-Bit und einem Stop-Bit umschlossen. Da das aber automatisch passiert und nicht zur eigentlichen Syntax gehört, werden wir das ignorieren.

Vorhang
auf!

Jetzt sehen wir uns eine MIDI-Nachricht an, mit der ein Ton eingeschaltet wird - den sogenannten Note-ON-Befehl:

Bit-Nummer	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1
Status-Byte	1	0	0	1	0	0	0	0
1. Daten-Byte	0	1	0	0	0	1	0	1
2. Daten-Byte	0	1	1	1	1	1	1	1

Das Status-Byte (10010000, bzw. 144) überträgt den Befehl **Note ON**. Da es sich um eine Channel-Nachricht handelt, steckt der eigentliche Befehl in den Bits 5-8 und der Kanal in den Bits 1-4. Letztere können die Werte 0-15 darstellen, was den Kanälen 1-16 entspricht.

Das erste Daten-Byte (01000101 bzw. 69) steht für den Kammerton a. Wie sich dieser anhört, hängt davon ab, welcher Klang/Sound bei dem empfangenden Gerät und der gewählten Kanalnummer gerade aktiv ist.

Das zweite Daten-Byte überträgt mit dem Wert 01111111 (bzw. 127), die Lautstärke des Tones. Genau genommen heißt dieser Wert "Velocity" und wäre bei einem Synthesizer die Anschlagstärke der verwendeten Taste.

Da keine Tondauer gespeichert wird, wird der Ton nach der gewünschten Zeit mit einem **Note OFF**-Befehl beendet. Dieser enthält das Status-Byte (mit dem korrekten Kanal), ein Daten-Byte mit der selben Tonhöhe wie beim Note-ON-Befehl (sonst bleibt der Ton an!) und ein Daten-Byte mit der "Loslass-Geschwindigkeit".

Ein
schweizer
Taschen-
Messer

Der SysEx-Befehl ist ein magisches Werkzeug, um all die Daten zu bearbeiten, die nichts mit Musik zu tun haben. Festgelegt sind hier nur die Statusbytes 240_{DEC} (Beginn) und 247_{DEC} (Ende). Alle dazwischen liegenden Bytes sind Datenbytes und herstellerspezifisch! Die ersten zwei Datenbytes können eine von der MMA vergebene Hersteller-ID beinhalten. Das vorletzte Byte kann eine Prüfsumme enthalten, die ebenfalls herstellerspezifisch aufgebaut ist.

In dem folgenden Beispiel wird ein Roland JD-990 Synthesizer angewiesen, den Effekttyp des aktuellen Klangprogrammes auf "Hall 4" zu ändern. Die zugehörige Sys-Ex-Sequenz lautet: 240, 65, 16, 87, 18, 1, 0, 0, 41, 5, 81, 247 (alle Werte Dezimal).

- Nach dem Status-Byte **240** folgen 3 Datenbytes, die den Empfänger spezifizieren: **65** für Roland, die Gerätenummer (in dem Fall **16**) und die **87** für den JD-990.
- Der Folgende Wert **18** teilt dem JD-990 mit, dass er Daten zu empfangen hat (Data Set). Ein Wert 17 würde Daten vom Gerät anfordern (Data Request)
- Nun folgen 4 Bytes, die festlegen, an welcher Speicheradresse ein Wert abgelegt werden soll: **1, 0, 0, 41**. Und schließlich der Wert selbst, der gespeichert werden soll: die **5** (der JD-990 akzeptiert an dieser Stelle Werte zwischen 0 und 9).
- Jetzt wir es etwas technisch, weil eine Prüfsumme folgt, die sich auf zwei Arten berechnen lässt:
 - Addition und Subtraktion: Die Bytes für die Adresse und die eigentliche Informationen werden addiert. Jedesmal, wenn der Wert 127 überschreitet, wird dieser um 128 reduziert. So wird die Prüfsumme nie größer als 127! Das Endergebnis wird schlussendlich von 128 subtrahiert.
 - Addition und Division: Diese Version ist etwas komplizierter, lässt sich aber in eine elegante Formel packen: O. g. Bytes werden addiert und anschließend durch 128 dividiert. Der Rest der Division wird von 128 subtrahiert: $128 - ((\text{Adresse1} + \text{Adresse2} + \text{Adresse3} + \text{Adresse4} + \text{Wert}) \text{ Mod } 128)$. Hier: $128 - ((1 + 0 + 0 + 41 + 5) \text{ Mod } 128) = 81$
- Die Meldung wird mit dem Statusbyte **247** abgeschlossen.

Zum Abschluss dieses Kapitels etwas exotisches für hartgesottene MIDI-Fans:

Innerhalb des herstellerspezifischen Bereiches bestehen auch Möglichkeiten, die Übertragung der Datenbytes individuell zu gestalten. Der Hersteller Korg tut das z. B., in dem er das für Daten-Bytes "verbotene" Bit 8 mit nutzt. Ein "Korg-Empfänger" rechnet immer mit Daten-Bytes, bei denen Bit 8 ebenfalls belegt ist. Für den Sender bedeutet das, immer ein Paket von 8 Bytes zu senden. Die Bytes 2-8 sind reguläre Daten-Bytes. Das erste Daten-Byte beinhaltet in den Bits 1-7 jeweils das Bit 8 der folgenden 7 Daten-Bytes. Der Korg-Empfänger bastelt sich also aus Byte1-Bit1 und Byte2 ein Daten-Byte, bei dem alle 8 Bits gefüllt sind. Aus Byte1-Bit2 und Byte3 das nächste Daten Byte, u.s.w. Das füllt zwar den Speicher optimal, kostet aber Zeit zum konvertieren. Und Zeit ist das wertvollste Gut bei der Übertragung: MIDI-Bytes werden seriell übertragen. Zu viele Daten und zu lange Leitungen können dafür sorgen, dass ein Akkord wie ein Arpeggio klingt ...

- Spezifikation
- Schaltung

Von
Volt
und
Ohm

Jedes Gerät, das über einen entsprechenden Anschluss und eine Logik zur Auswertung (Software) verfügt, kann MIDI-Daten senden und empfangen. Da MIDI in erster Linie ein Datenprotokoll ist, ist die verwendete Hardware sekundär.

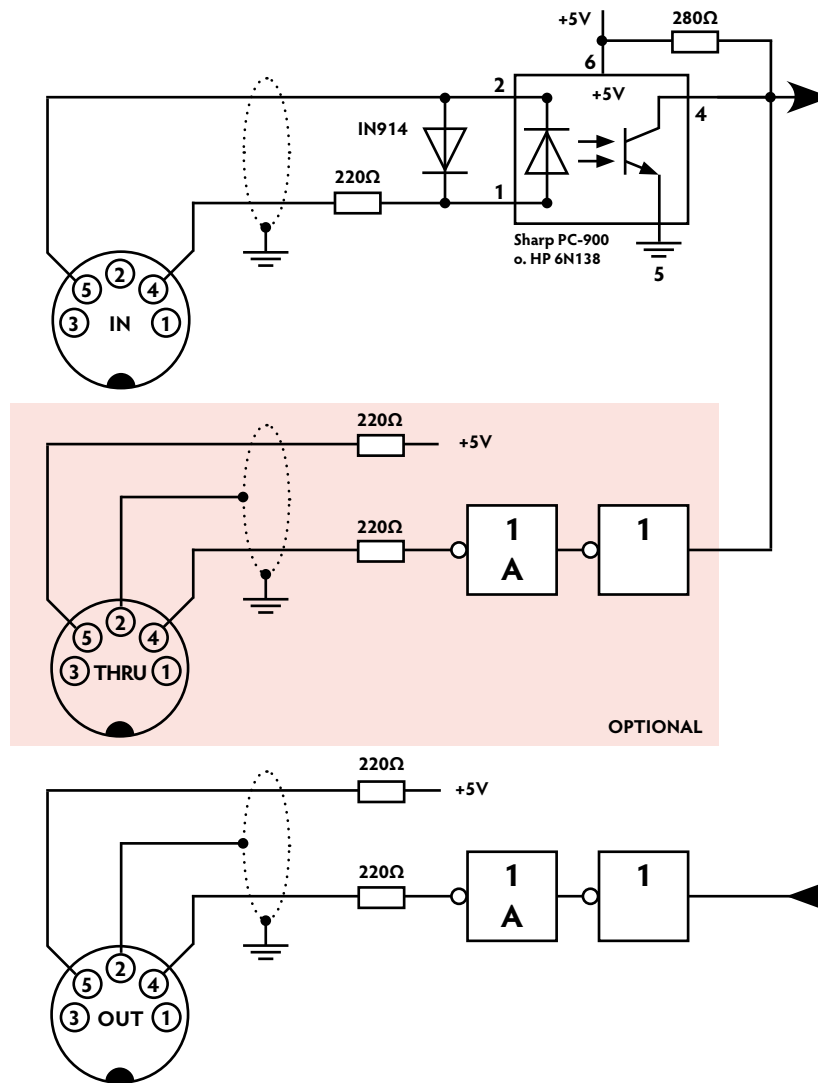
Klassischerweise* verwenden MIDI-Geräte drei 5-polige DIN-Buchsen (DIN5/180°), auch Diodenbuchsen genannt. Mehrere Anschlüsse sind deshalb erforderlich, weil MIDI unidirektional, also nur in einer Richtung arbeitet. Es werden somit mindestens ein MIDI-In- und ein MIDI-Out-Port benötigt. Ein optionaler MIDI-Thru-Port ermöglicht es, das Signal, das am In-Port anliegt, durchzuschleifen.



Die Pins 4 und 5 dienen als Datenleitungen, wobei Pin 4 über 220Ω mit +5V und Pin 5 über 220Ω mit 0V verbunden ist (s. folgende Seite). Strom wird als „logisch 0“, kein Strom als „logisch 1“ definiert. Ein Optokoppler in der Empfangsleitung sorgt für eine galvanische Trennung. Die Masse (Pin 2 und Abschirmung) wird nicht angeschlossen, um Masseschleifen zu vermeiden.

Die Übertragungsgeschwindigkeit beträgt 31250 Bit/s, was exakt $32\ \mu\text{s}$ pro Bit oder oder $960\ \mu\text{s}$ für 3 Bytes (1 MIDI-Befehl) bedeutet. Damit sind der Datenmenge und dem Timing klare Grenzen gesetzt. Es sollte deshalb vermieden werden, mehr als 3 Geräte in Reihe zu schalten, bzw. eine MIDI-Verbindung (mit oder ohne zwischengeschaltete Geräte) länger als 15 Meter werden zu lassen.

* Heute werden MIDI-Daten auch über USB-, Firewire- oder WLAN-Schnittstellen übertragen. Diese Schnittstellen bieten ein deutlich besseres Timing und heben auch einige andere Einschränkungen auf. So legt die MIDI-Spezifikation für USB z. B. fest, dass jeder der 16 USB-Endpunkte (Jacks) weitere 16 Kanäle enthält.

Licht
und
Strom

So sieht die Original-Schaltung aus den 80ern aus (räumlich optimiert und die ANSI-Symbole gegen DIN-Symbole getauscht).

Diese Schaltung hat eigentlich nur noch nostalgischen Wert, da man MIDI-Schnittstellen, Adapter von MIDI auf USB, MIDI-Patchbays und vieles mehr, heute in großer Auswahl und für wenig Geld bekommt.

Wer Spaß am Löten hat, findet im Internet entweder "klassische" Anleitungen, oder kann solches auf der Basis eines Raspberry-Pi*- bzw. eines Arduino-Systemes* umsetzen.

Für die nebenstehende Schaltung gilt:

Die Widerstände müssen über eine Toleranz von 5% verfügen. Der Farbcode für 220Ω ist demnach ■■■■■ und für 280Ω ■■■■■.

Die beiden Nicht-Gatter, die in Reihe geschaltet das Signal verstärken, können als Transistoren aufgebaut werden. Die Nicht-Gatter mit der Bezeichnung A dürfen wahlweise auch als IC verbaut werden.

* Beides sind komplette Computer im Scheckkartenformat, deren Hard- und Software Open-Source ist. Damit lassen sich mit wenig Aufwand beliebige Computerprojekte entwickeln. Gedacht sind diese Low-Cost-Systeme, um den Erwerb von Hard- und Softwarekenntnissen zu erleichtern. Ein Stück weit wird damit die verlorengegangene Heimcomputer-Kultur der 80er Jahre wiederbelebt.

- Damals
- Heute



Es
war
einmal
...

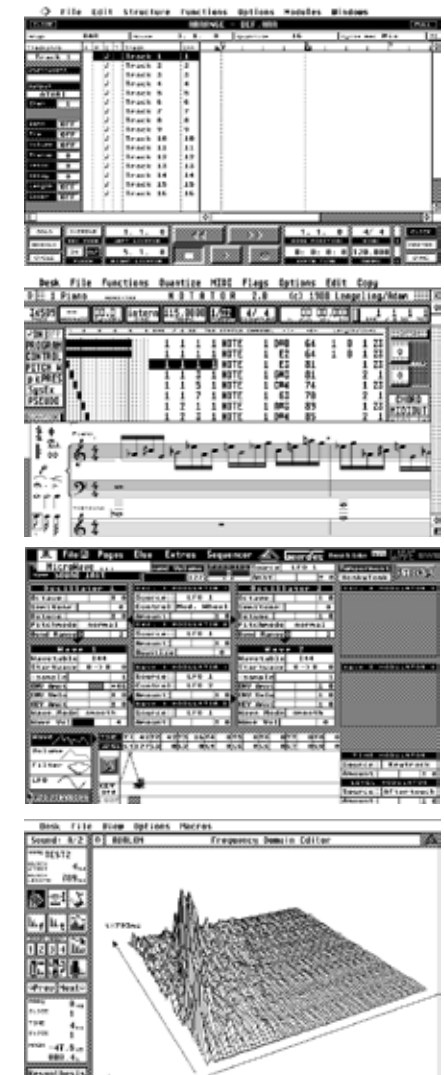
Mit der Erscheinung von MIDI fand sich bald auch allerlei MIDI-taugliche Software auf den damals verfügbaren Computer-Plattformen ein (C64, Macintosh, Atari, PC).

Zu den Standardanwendungen gehörte der **Sequencer**, mit dessen Hilfe MIDI-Nachrichten in einem definierten Zeitraster aufgenommen wurden. Waren die MIDI-Daten einmal im Computer, konnten sie als Noten, Balken oder Zahlen dargestellt und einfach bearbeitet werden. Per MIDI angeschlossene Instrumente übernahmen die Tonerzeugung. Die bekanntesten Vertreter ihrer Art waren Notator (C-Lab, später Emagic) und Cubase (Steinberg).

Eine weiterer Anwendungsbereich waren **Sound-Editoren**, mit denen die Klangdaten angeschlossener Synthesizer bearbeitet und organisiert werden konnten. Emagic bot hierzu eine besondere Lösung (Polyframe). Diese stellte ein Framework zur Verfügung, mit dem sich mehrere Editoren als zukaufbare Module unter einer Oberfläche verwalten ließen. Ein sogenanntes Universal-Modul erlaubte darüber hinaus das Erstellen eigener Editoren.

Etwas exotischer waren **Audio-Editoren**. Bekanntester Vertreter dieser Zunft war Steinbergs Avalon, mit dem sich Audio-Dateien diverser Sampler (z.B. Akai S1000, Roland S50, Ensoniq EPS, etc.) laden, bearbeiten und wieder zurückspeichern ließen.

Die hier gezeigten Screenshots stammen von Atari-Computern mit einer Auflösung von 640x400 Bildpunkten (schwarz/weiß). Atari-Computer boten eine ähnlich komfortable Oberfläche, wie der Macintosh - allerdings zu einem deutlich niedrigeren Preis. Ein großer Pluspunkt des Atari war die eingebaute MIDI-Schnittstelle, die diese Plattform sehr schnell zum Standard in der Musik-Szene machten.



Alles
unter
einem
Dach



Logic Pro X ist eine aktuelle Anwendung, die alle zuvor genannten Aspekte unter einer Oberfläche integriert: Sequencer und Editoren für MIDI- und Audio-Daten, virtuelle Instrumente und Musiker (Gitarristen, Schlagzeuger), Synchronisation von Videoclips, professioneller Notensatz, ein programmierbares MIDI-Environment, Mischpult und Effektgeräte. Ich hoffe, ich habe nichts vergessen...

- Status-Bytes
- Control-Change

Die
Vokabeln
der
MIDI
Sprache

Es folgt eine Übersicht über die MIDI-Befehle, aufsteigend sortiert nach ihren Status-Bytes (binär)

Status (Bin)	Dec	Hex	Befehl	Daten 1 (Dec)	Daten 2 (Dec)	Notiz
1000cccc	128	8c	Noten Off	(Tonhöhe) 0-127	(Release) 0-127	Interessant z. B. für Bläser-Sounds
1001cccc	144	9c	Note On	(Tonhöhe) 0-127	(Velocity) 0-127	
1010cccc	160	Ac	Key-Pressure (Aftertouch polyphon)	(Tonhöhe) 0-127	(Wert) 0-127	z.B. Klangveränderungen durch nachträglichen Druck
1011cccc	176	Bc	Control Change	(Controller) 0-127	(Wert) 0-127	Detaillierte Tabelle auf der nächsten Seite
1100cccc	192	Cc	Programm Change	(Wert) 0-127		
1101cccc	208	Dc	Channel-Pressure (Aftertouch)	(Wert) 0-127		
1110cccc	224	Ec	Pitch Bending	(Wert) 0-127	(Wert) 0-127	Zwei Datenbytes für höhere Auflösung (65535 Werte)
11110000	240	F0	System Exclusive			Erläuterungen auf Seite 11
11110010	242	F2	Song Position Pointer	(Wert) 0-127		SequencerSteuerung
11110011	243	F3	Song Select	(Wert) 0-127		SequencerSteuerung
11110110	246	F6	Tune Request			für analoge Synthesizer
11110111	247	F7	End of Exclusive			Erläuterungen auf Seite 11
11110000	248	F8	Timing Clock			SequencerSteuerung (Sync: 24 x pro definiertem Grundschatz gesendet)
11110100	250	FA	Start			SequencerSteuerung
11110101	251	FB	Continue			SequencerSteuerung
11111100	252	FC	Stop			Sequencer-Steuerung
11111110	254	FE	Active Sensing			"Totmann-Schaltung", alle 300ms
11111111	255	FF	System Reset			Power-Up-Conditions

Alle Status-Bytes, deren letzten 4 Bits "cccc" lauten, sind Channel-Messages, übertragen also zusätzlich einen MIDI-Kanal. Control-Change und System Exclusive (SysEx) sind etwas komplexer. Die Control-Change-Optionen werden deshalb auf der nächsten Seite erläutert. Der Sys-Ex-Befehl wurde bereits auf Seite 12 besprochen. Sowohl in dieser, als auch in der folgenden Tabelle kann man erkennen, dass nicht alle der möglichen Status-, bzw. Daten-Bytes Verwendung finden. Das lässt Spielraum für künftige Erweiterungen.

Ein Sonderfall ist der Running-Status. Werden z. B. mehrer Note On Meldungen gesendet, genügt es, zu Beginn das Status-Byte einmalig zu übertragen. Ein entsprechend befähigter Empfänger wird erkennen, dass alle folgenden Datenbytes zum Note-On-Status gehören; und zwar so lange, bis ein weiteres Status-Byte empfangen wird. Das funktioniert aber nicht immer unfallfrei ;-)

Kontrolle
ist
alles!

Und hier alle Möglichkeiten des Control-Change-Befehles

Status (Bin)	Dec	Hex	Daten 1 (Dec)	Befehl	Daten 2 (Dec)	Notiz
1011cccc	176	B0	0	Bank Select MSB		Herstellerabhängig
1011cccc	176	B0	1	Modulation	(Wert) 0-127	
1011cccc	176	B0	2	Breath Control	(Wert) 0-127	
1011cccc	176	B0	4	Foot Control	(Wert) 0-127	
1011cccc	176	B0	5	Portamento Time	(Wert) 0-127	
1011cccc	176	B0	6	Data Entry	(Wert) 0-127	
1011cccc	176	B0	7	Main Volume	(Wert) 0-127	
1011cccc	176	B0	8	Balance	(Wert) 0-127	
1011cccc	176	B0	10	Panorama	(Wert) 0-127	
1011cccc	176	B0	11	Expression	(Wert) 0-127	
1011cccc	176	B0	32	Bank Select LSB		Herstellerabhängig
1011cccc	176	B0	64	Sustain (Pedal)	0 (off), 127 (on)	
1011cccc	176	B0	65	Portamento	0 (off), 127 (on)	
1011cccc	176	B0	66	Sostenuto (Pedal)	0 (off), 127 (on)	
1011cccc	176	B0	67	Soft (Pedal)	0 (off), 127 (on)	
1011cccc	176	B0	91	Extern EFX depth	(Wert) 0-127	
1011cccc	176	B0	92	Tremolo depth	(Wert) 0-127	
1011cccc	176	B0	93	Chorus depht	(Wert) 0-127	
1011cccc	176	B0	94	Detune depth	(Wert) 0-127	
1011cccc	176	B0	95	Phaser depth	(Wert) 0-127	
1011cccc	176	B0	96	Data increment	(Wert) 0-127	
1011cccc	176	B0	97	Data decrement	(Wert) 0-127	
1011cccc	176	B0	121	Reset all Controller	0	
1011cccc	176	B0	122	Local Control	0 (off), 127 (on)	
1011cccc	176	B0	123	All Notes off	0	
1011cccc	176	B0	124	Omni Off	0	All Notes off
1011cccc	176	B0	125	Omni On	0	all Notes off
1011cccc	176	B0	126	Mono Mode	(Kanäle) 1-16	Poly Mode off, all Notes off
1011cccc	176	B0	127	Poly Mode	0	MonoMode off, all Notes off

Der Control-Chance-Befehl ist eines der Status-Bytes aus der vorangegangenen Tabelle. Er besagt zunächst einmal, dass ein Controller zu Anwendung kommen soll (wenn gesendet), bzw. ein Controller benutzt wurde (wenn empfangen).

Das erste Datenbyte spezifiziert den Controller, das zweite Datenbyte den zugehörigen Wert.

So kennt der Controller 7 (Main Volume) z.B. 127 verschiedene Lautstärken und Controller 64 (Sustain - oder Haltepedal) die Werte 127 für gedrückt und 0 für losgelassen.

• Quo Vadis?



Future
will
be
exciting!

Nun ist MIDI seit über 30 Jahren im Einsatz. In dieser Zeit ist eine Menge passiert: Zum einen wurden Computer um ein vielfaches leistungsfähiger, zum anderen sind heute Geräte in der Lage mit MIDI umzugehen, mit denen damals noch niemand gerechnet hätte. So z.B. Smartphones und Tablets.

Außerdem kommt MIDI in auch anderen Bereichen zum Einsatz, bei dem es um steuern und regeln geht. So z.B. in der Bühnen- und Lichttechnik. Ein anderes Beispiel ist ReWire. Ein Protokoll das MIDI benutzt, um Steuerungs- und Audiodaten in Echtzeit zwischen zwei Programmen auszutauschen, bzw. Audioanwendungen synchron laufen zu lassen.

Jetzt steht MIDI 2.0 in den Startlöchern. Die gute Nachricht vorweg: MIDI 2.0 ist komplett abwärtskompatibel. Somit bleiben uns alle lieb gewordenen MIDI-fähigen Raritäten, die 5poligen Diodenstecker und USB erhalten. Kommende MIDI-Hardware wird also miteinander kommunizieren, um festzustellen, welches Protokoll zum Einsatz kommt.

Da MIDI 2.0 bidirektional arbeitet, wird die MIDI-Thru-Buchse wohl überflüssig werden. Das parallele Weiterleiten einer MIDI-Unterhaltung zwischen zwei Geräten ist wohl auch eher problematisch als hilfreich.

Was wird sich mit MIDI 2.0 konkret ändern?

- Die ursprünglich vermuteten "tausende Kanäle" haben sich auf 256 eingependelt, die in 16 Gruppen zu je 16 Kanälen verwaltet werden. Eine Gruppe kann entweder MIDI 1.0 oder MIDI 2.0 "sprechen".
- MIDI 2.0 wird über eine umfassende Artikulations-Steuerung verfügen. 256 Artikulationstypen (Spielweisen, jeweils 8Bit) und deren zugehörige Daten werden in 256 Abstufungen übertragen.
- Des Weiteren führt MIDI 2.0, die aus VST3 bekannten MIDI-Controller-Daten pro Note ein. Die Auflösung wird 32Bit betragen.
- Pitchbend und Anschlagsdynamik müssen nun nicht mehr mit 7+7Bit übertragen, sondern haben gleich 16Bit zur Verfügung.
- Die Auflösung der Controller beträgt jetzt 32Bit.
- Dass die Kommunikation bidirektional erfolgt, hatte ich ja schon erwähnt.

Man darf also gespannt sein, welche neuen Möglichkeiten sich damit eröffnen.

- Print & Web
- Software

horizont-
erweiternd**Basics:**

- MIDI 1.0 Detailed Specification - Document Version 4.1.1, the International MIDI Association, Los Angeles, 1990
- Midi Manufacturers Association: <http://www.midi.org/techspecs/> *
- [§] Fuera/Scacciaferro, The MIDI Impementation Book, Third Earth Publishing Inc., Pompton Lakes , 1986, ISBN 0-88188-558-4
Interessant für Anwender von MIDI-Technik vor 1987

Einführungen:

- [§] Enders/Klemme, Das MIDI und SOUND-Buch zum Atari ST, Markt&Technik Verlag AG, München, 1988, ISBN 3-89090-528-5
Ab Seite 81 eine ganz hervorragende Einführung in die Materie. Incl. Anleitungen zum erstellen von eigenen MIDI-Anwendungen via GfA-Basic
- [§] Hofmann/Prischl/Sasso, Das große Logic-Audio-Handbuch, CG Carstensen Verlag, München, 1999, ISBN 3-910098-15-0
Der 654-seiten schwere Wälzer enthält auf über 100 Seiten detaillierte Infos und Anleitungen zum nutzen des Logic-MIDI-Environment - der MIDI-Spielwiese schlechthin

[§] Bücher mit diesem Vermerk sind nur noch antiquarisch zu bekommen. Ein guter Start für eine Recherche ist das www.zvab.de

* Links lassen sich in einem PDF-Dokument anklicken. Das Linkziel wird in Ihrem Webbrowser angezeigt.

Werkzeug.

Sequencer/Produktionsumgebungen:

- Logic Pro X: <http://www.apple.com/de/logic-pro/> [Mac]
- Avid Protools: <http://www.avid.com/de/products/family/pro-tools> [Mac][Win]
- Cubase: <https://www.steinberg.net/de/products/cubase/start.html> [Mac][Win]
- Ableton Live: <https://www.ableton.com/de/live/> [Mac][Win]
- Propellerhead Reason: <https://www.propellerheads.se/reason> [Mac][Win]

Softwareinstrumente/PlugIns/Effekte: *

- Native Instruments: <http://www.native-instruments.com/de>
- Izotope: <https://www.izotope.com>
- Waldorf: <http://www.waldorf-music.info/de>
- Output: <http://output.com>
- Softube: <https://www.softube.com>
- Arturia: <https://www.arturia.com>

Notensatz:

- Sibelius: <http://www.avid.com/DE/products/sibelius> [Mac][Win]
- Finale: <http://www.klemm-music.de/notation/index.php> [Mac][Win]
- Capella: <http://www.capella.de/de/index.cfm/produkte/capella/info-capella/> [Win]

Entwicklungsumgebungen:

- Cycling '74 Max: <https://cycling74.com> [Mac][Win]

* Die unter Sequenzer/Produktionsumgebungen aufgeführten Anwendungen enthalten zwar schon jede Menge Instrumente, die aber größtenteils den Mainstreambereich abdecken. Diese Angebote bieten ein paar ungewöhnliche Erweiterungen, die alle einen genaueren Blick wert sind.

- Was ist das?
- Projekte

Do
it
your
self

Mit MIDI kann man allerhand anstellen. Z. B. auch eigenes Equipment bauen, um damit MIDI-fähige Geräte anzusteuern. Besonders beliebt: Controller-Boxen mit Tastern, Fadern und Drehregler. Aber auch Klangerzeugende Elektronik ist keine Sache der Unmöglichkeit. Voraussetzung ist eine gewisse Grundfertigkeit im Umgang mit Lötkolben und elektronischen Bauteilen, das sich ggf. auch erlernen lässt.

Was die Verarbeitung der Daten angeht, kann man auf hersteller-spezifische Lösungen zurückgreifen, oder sich mit einer universellen Kleinstrechner-Lösung der Gattung Arduino oder Rapsberry PI anfreunden.

Im Netz kursieren eine Menge Portale mit Informationen, Anleitungen, Filmen und Material zu DIY-Projekten. Auf der folgenden Seite werden ein paar handverlesene Beispiele* gezeigt.

* Veraltete oder fehlerhafte Links dürfen gerne unter info@be-sign.net gemeldet werden.

...
und
so
wirds
gemacht

- **MakeProAudio** (<https://www.makeproaudio.com>): Das MakeProAudio-Projekt hat seine Wurzeln in dem legendären Unternehmen "Creamware", das Anfang der 90er begann, DSP-Hardware herzustellen. Laut eigenen Angaben hat das Unternehmen ist in der Nähe von Köln angesiedelt und verfügt über ein Forschungs- und Entwicklungszentrum in Bangalore (Indien), und Teammitglieder rund um den Globus. Im April 2019 hatte ich während der Musikmesse 2019 die Gelegenheit, das damals neue Projekt näher kennen zu lernen. Fazit: Persönliche Empfehlung ✓
- **Doepfer** (<http://www.doepfer.de>): Der Klassiker auf dem Markt des Musikelektronik-Eigenbaus. Alles begann im Oberbayrischen Gräfelfing, als Dieter Döpfer 1977 einen Voltage Controlled Phaser (VCP) für den Formant, einen analogen Selbstbausynthesizer aus der Zeitschrift Elektor, entwickelte. Heute ist das Produktangebot riesig, beinhaltet aber so ziemlich alles um das Thema Musikelektronik. Legendär ist Döpfers modularer Synthesizer A-100. Zu den Nutzern von Döpfers Produkten zählen unter anderem Kraftwerk, Radiohead, Klaus Schulze, Daft Punk und einige andere mehr.
- **uCApps** (<http://www.ucapps.de>): uCApps ist ein nicht-kommerzielles Projekt von Thorsten Klose. Ausser den MIDIbox-Projekten, finden sich auf der Website auch einige User-Projekte, ein WIKI und ein Forum.
- **dartmobo** (<https://dartmobo.com>): Auf dieser, in Italien beheimateten Seite, findet man Hardware, Software und Dokumentationen zu Arduino-basierten MIDI-Controllern.
- **BASTL-INSTRUMENTS** (<https://bastl-instruments.com>): Diese eher ungewöhnliche Website von Václav Peloušek und Ondřej Merta aus der Tschechischen Republik, bietet eine große Auswahl an Modulen, Kits und Workshops rund um das DIY-Thema.
- **midi-flutes** (<https://www.midi-flutes.com/de>): Hier finden sich ein paar Selbstbau-Projekte für MIDI-Blaswandler. Diese sind ja nicht ganz billig, wenn man Sie von Professionellen Anbietern bezieht. aber es geht eben auch im DIY-Verfahren.
- **hackster** (<https://www.hackster.io/projects/tags/midi>): Zum Abschluss noch eine Sammlung von 53 sehr unterschiedlichen DIY-MIDI-Projekten, die als Anregung zu eigenen Ideen dienen können...

About
me...

Baujahr 1965, Südschwarzwald

Creativ-Worker. Print- & Webdesign (be-sign.net) und Dozent an zwei Musikschulen.

Hobbies: Fischertechnik, Retro-Computing, Science-Fiction, Musik, Bücher, Kochen, Architektur.

www.bernd-scheurer.de



be-sign.net ist ein Dienstleistungsunternehmen, das seit 2003 in den Bereichen Print- und Webdesign tätig ist.

Das Spektrum der Kunden reicht von Bildungs- und Sozialeinrichtungen, über Verlage und Autoren bis hin zu Forschung und Handwerk.

www.be-sign.net



Seit 1969 mit dem Klavier und seit Ende der 70er mit dem Computer zugange. Seit 1989 Erfahrung im Unterrichten von Schülern zwischen 5 und 75.

Themenschwerpunkte: Klavier, Keyboard, Musiktheorie und Musiksoftware/MIDI.

unterricht.bernd-scheurer.de

Team-
-Work

Workshops sind Team-Arbeit. be-sign.net kooperiert bei der Umsetzung mit verschiedenen Partnern, die die Angebote perfekt ergänzen. Aktuell sind folgende Partner mit an Bord:

Dr. Erkens

CONSULTING GROUP

consulting
management
communication

Die Dr. Erkens Consulting Group ist eine im Rhein-Main-Gebiet ansässige, inhabergeführte Beratungs- und Management-Agentur im Healthcarebereich mit den Dienstleistungsschwerpunkten Unternehmensberatung kleiner und mittelständiger Unternehmen, Managementdienstleistungen, Marketing- und Vertriebsstrategien und Produktkommunikation.

Zu ihren Kunden zählen internationale Pharma- und Medizintechnikunternehmen, Medizinverlage, Dienstleistungsunternehmen sowie Institutionen, Verbände, Gesellschaften und Vereine des Gesundheitsbereiches.

Dr. Erkens-Consulting und be-sign.net kooperieren bei Workshops mit Schwerpunkt Grafik-/Webdesign und Marketing.

www.dr-erkens-consulting.de



Die JMS-Erzhausen ist ein privates Musikinstitut mit Sitz in Erzhausen bei Darmstadt. Ein Team von unabhängigen und selbständigen Musiklehrern unterrichtet unter einem Dach, um ihnen die größtmögliche Auswahl an musikalischen Möglichkeiten zu bieten.

Das Angebot der Lehrkräfte reicht von den Anfängen der musikalischen Frühförderung bis zum Erwachsenenunterricht. Ob Einzel- oder Gruppenunterricht, Kammermusik oder Bandworkshops, Seminare oder Einzelförderung, für jeden steht das passende Angebot zur Verfügung.

Die JMS-Erzhausen kooperiert mit be-sign.net bei Workshops mit Schwerpunkt Musik und -Software.

www.jms-erzhausen.de

Was
gehört
wem?

Soweit nicht anders erwähnt, sind alle verwendeten Fotos ©by be-sign.net.

- Portraitfoto Dave Smith: ©East Bay Express, Oakland
- Portraitfoto Ikutaro Kakehashi: ©by heise
- Kapitelfoto Zukunftsmusik: ©by NASA

Ihre
Notizen



A series of 20 horizontal lines for writing notes, arranged in a single column.

be-sign.net
bernhard r. scheurer
mainstrasse 2 • 64390 erzhausen
www.be-sign.net • info@be-sign.net

