

MIDI

Walter Riemer

In diesem Aufsatz wird in erster Linie auf die Grundlagen von MIDI eingegangen. Das direkte Zusammenwirken von MIDI-Instrumenten und -Geräten bleibt weitgehend außer Betracht, da es in erster Linie für ausübende Musiker von Interesse ist. Ziel ist vor allem, den technischen Hintergrund zu erklären, wenn auch die komfortablen Sequencer-Programme die technischen Details in den Hintergrund treten lassen. Trotzdem hilft es sehr, wenn man Bescheid weiß, was in diesem Hintergrund vor sich geht.

Eine ausführliche Darstellung der MIDI-Spezifikation befindet sich auf <http://vision1.cs.UMR.edu/~johns/links/music/midispec.htm>.

1 Allgemeines und Grundsätzliches über die Hardware

MIDI ist die Abkürzung für **M**usical **I**nstrument **D**igital **I**nterface. Eine Gruppe von Herstellern, die MIDI Manufacturers Association, gab 1983 die MIDI Specification 1.0 heraus, die trotz Vorhandenseins einer erweiterten Fassung nach wie vor gültig ist.

Ursprünglich wurde MIDI konzipiert, um das Zusammenwirken mehrerer MIDI-Instrumente bzw. -Geräte zu standardisieren.

Heute (etwa seit Anfang der 80er-Jahre) ist MIDI auch die Antwort auf das Problem, dass digital aufgenommene Musik sehr speicheraufwendig ist und für die Bearbeitung und Weiterverarbeitung einen sehr hohen Aufwand erfordert (hinsichtlich CPU-Leistung, Festplattengröße, aber auch Software), jedenfalls viel mehr als Text- oder Bildbearbeitung. Die erforderliche Software erfüllt (mit entsprechend leistungsfähigen Computern) heute schon höchste Ansprüche.

MIDI basiert auf den Konzepten verteilter Intelligenz und Echtzeitverarbeitung und dem Wunsch, eine möglichst einfache Sprache zum Beschreiben klingender Musik zu haben. Ziel war, eine einheitliche Spezifikation von Hardware und Software zu schaffen, nach der (elektronische) Instrumente ferngesteuert werden können ("remote control"), sodass Geräte unterschiedlicher Hersteller zu einem persönlichen Musikstudio kombiniert werden können. Praktisch alle elektronischen Instrumente sowie viele bessere Audiogeräte sind heute mit MIDI ausgestattet.

2 MIDI-Hardware

2.1 Klangerzeugung (Synthesizer)

MIDI ist ein **Steuerungsprotokoll**, mit dem Grundparameter von Klängen, nicht aber die Klänge selbst übermittelt werden; dadurch bleibt der Umfang zu übertragender bzw. abzuspeichernder Daten relativ gering.

Die Klänge müssen von einer durch MIDI-Daten gesteuerten Einrichtung erzeugt werden.

Ein **Synthesizer** ist ein solches Gerät zum synthetischen Erzeugen von Klängen (Sounds). Diese können konventionellen Instrumenten entsprechen, aber auch völlig neuartig sein.

In **FM-Synthesizern** werden die Klänge rein synthetisch (durch Mischen mehrerer Outputs von Frequenzgeneratoren) erzeugt. Diese Mischung kann rein additiv erfolgen (Parallelschaltung der Generatoren), aber auch, indem die Grundschiwingung als Trägerfrequenz moduliert wird (Serienschaltung der Generatoren). Die Generatoren bestehen aus Oszillator, Hüllkurvengenerator und Level-Controller und heißen in dieser Gesamtheit Operatoren.

In **PCM-Synthesizern** werden digitalisierte Klänge echter Instrumente verwendet, die mittels A/D-Wandlers digitalisiert wurden. Die Klangqualität ist je nach Hardwareaufwand (insbesondere Speicherkapazität) wesentlich besser.

Sound-Karten enthalten Kombinationen von Chips aus den Bereichen digitaler Messtechnik (DSP - *Digital Signal Processing*) einerseits und elektronischer Musikinstrumente andererseits, daher von Herstellern wie *National Semiconductors* einerseits und *Yamaha* und *Roland* andererseits.

Sound-Karten enthalten im wesentlichen folgende Komponenten:

- A/D- und D/A-Wandler (CODEC = Codierer/Decodierer)
- Synthesizer
- MIDI-Schnittstelle

Dem Verkehr zwischen der Sound-Karte und der Software unter Windows dienen die **Windows Software-Interfaces**:

- Media Control Interface (MCI)
- API (für Low-Level Systemaufrufe)

Die Klangfarben beziehungsweise deren Zuordnung zu bestimmten Nummern stehen im Prinzip den Herstellern frei, jedoch

wurde zunächst von der Firma Roland das General MIDI (GM) - System festgelegt und aus Kompatibilitätsgründen von vielen Herstellern übernommen. Es stehen 128 Klangfarben zur Verfügung, welche außer konventionellen Instrumenten auch Geräusche (zum Beispiel Wind, Automotor u.dgl.) enthalten. Die internen Nummern dieser Klangfarben sind 0 bis 127; MIDI-Geräte und MIDI-Software zeigen jedoch häufig Nummern von 1 bis 128 an. Die Klangfarben werden in der MIDI-Fachsprache mit *Program* oder *Patch* bezeichnet.

Die Klangfarben (Programs, Patches) sind in Achtergruppen gegliedert:

1-8	Klaviere,
9-16	Metall-Schlaginstrumente,
17-24	Orgeln,
25-32	Gitarren,
33-40	Bass,
41-48	Streichinstrumente,
49-56	Ensembles,
57-64	Blech,
65-72	Rohrblatt,
73-80	Flöten u.dgl., usw.,
105-112	Volksinstrumente, usw.

Obwohl eine Gruppe von Schlaginstrumenten enthalten ist, sind darüber hinaus noch wesentlich mehr mit den Nummern 35 bis 81 (intern 34 bis 80) festgelegt. Diese werden automatisch mit Kanal 10 angesprochen.

Polyphonie ist die Fähigkeit, in einem Kanal mehrere Noten (aber gleicher Klangfarbe) gleichzeitig zu erzeugen: ein 32fach polyphoner (32-stimmiger) Synthesizer kann dementsprechend bis zu 32 Noten in einem Kanal gleichzeitig erzeugen.

Multitimbralität beschreibt die Fähigkeit eines Synthesizers, eine bestimmte Anzahl von Klängen (Klangfarben, Instrumenten) gleichzeitig zu erzeugen, ist also gleichzusetzen der Anzahl voneinander unabhängiger Klangerzeuger ("timbre" = Klangfarbe).

Erweiterungen der GM-Norm wurden von den Firmen Roland selbst (GS) und Yamaha (XG) vorgenommen; GS ist jedoch weiter verbreitet als XG.

2.2 Sequencer

Ein **Sequencer** kann **MIDI-Events** (Ereignisse, beschrieben durch bestimmte Parameter, also etwa eine bestimmte

Note zu einem bestimmten Zeitpunkt mit bestimmter Klangfarbe und anderen Attributen) im korrekten Zeitmaß aufnehmen oder abspielen.

2.3 Verbindungen zwischen MIDI-Geräten



Verbindungen werden mittels 5-poliger DIN-Steckvorrichtungen hergestellt (Buchse am Gerät, Stecker an den Verbindungskabeln), wobei Eingang und Ausgang an getrennten Steckvorrichtungen liegen. Viele Geräte haben auch eine Durchgangsbuchse (MIDI-Thru), an der das Eingangssignal unverändert wieder ausgegeben wird.

Die MIDI-Eingänge sind mit Optokopplern zwecks galvanischer Trennung der Geräte ausgestattet.

Laut Spezifikation dürfen die Kabel maximal 50 Fuß (rund 17 m) lang sein; aktive Adern (für "Hin- und Rückleitung der elektrischen Signale") sind 4 und 5, als "twisted pair" paarweise geschirmt; der Schirm liegt auf 2; 1 und 3 werden nicht benützt:

Die Serienschaltung ("Daisy-chaining") von Geräten ("Devices") mittels MIDI-Thru ist jedoch begrenzt, da jeder Optokoppler eine gewisse Verzögerungszeit (von höchstens 2 µs) hinzufügt.

Die Verbindung arbeitet als asynchrone serielle Schnittstelle mit einer Baudrate von 31.25 kBaud (±1%), 8 Datenbits, 1 Startbit und 1 Stopbit; die Übertragung eines Bytes benötigt demnach 320 µs.

Das Funktionsprinzip ist das einer Stromschleife mit 5 mA. Logisch 0 ist fließender Strom.

2.4 MIDI-Messages

MIDI-Messages werden von einem Sender (Master) abgegeben und haben die Aufgabe, den Empfänger (Slave) zu steuern. Ein Großteil der MIDI-Messages beschreibt **Ereignisse** ("Events") im Zuge eines Musikstücks, zum Beispiel das Ein- oder Ausschalten einer Note. Vor dem Ausschalten steht in der Regel ein **Delta-Time**-Wert, der die Zeit bis zum Eintreten dieses Events bestimmt.

An Events gibt es drei Arten:

MIDI-Event, eventuell begrenzt mit F7h: jede Message, die einen bestimmten Kanal betrifft.

System-Exclusive-Event, eingeleitet mit F0h, begrenzt mit F7h: ermöglicht ähnlich ESCAPE, gerätespezifische Messages zu senden, ohne den sonstigen Dateiinhalt zu beeinflussen.

Meta-Event, eingeleitet mit FFh: Informationen allgemeinen Charakters.

MIDI-Geräte werden gesteuert, indem ein Gerät an ein anderes MIDI-Messages sendet. Standardmäßige MIDI-Messages zeigen deutlich die Konzeption, dass ein Gerät als Sender ("Master"), das andere als Empfänger ("Slave") fungiert.

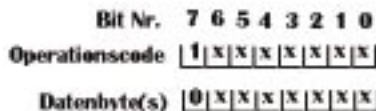
Die Form jeder Message ist

Jede MIDI-Message enthält zunächst ein Byte Operationscode (genannt Statusbyte), und anschließend meist ein bis mehrere (bis viele) Datenbytes.

Datenbytes haben demnach Inhalte bis einschließlich 127, Statusbytes darüber. Statusbytes haben so lange Gültigkeit, als nicht ein anderes Statusbyte gesendet wird.

Es gibt keine ACKs (Acknowledge) oder andere Handshaking-Mechanismen im MIDI-Protokoll. Eingestreuete 00h nach Messages sind ohne inhaltliche Bedeutung.

(1) Statusbytes



Statusbytes haben auf Bit 7 den Wert 1, liegen also im Wertebereich 80h bis FFh (numerische Werte von 128 bis 255); Datenbytes haben auf Bit 7 den Wert 0 und liegen daher im Wertebereich 0 bis 7Fh (numerische Werte von 0 bis 127). Im allgemeinen gilt jedes Statusbyte so lange, bis ein neues kommt. (Details siehe **Tabelle 1**)

(1.1) Statusbytes der Kategorie Voice

Statusbytes unter F0h (240d) gehören in die Kategorie Voice ("Stimme"), das heißt, die zugehörige Message kann über einen MIDI-Kanal gesendet werden und beschreibt damit den Inhalt einer Stimme (etwa eines auf mehrere Instrumente aufgeteilten Musikstücks). Jeder der 16 möglichen MIDI-Kanäle (intern nummeriert von 0 bis Fh = 15d, extern, das heißt auf den Geräten, jedoch von 1 bis 16) ist einer Stimme zugeordnet.

Diese Statusbytes können in zwei Halbbytes ("Nibbles") zerlegt gedacht werden.

- 8 = Note Off
- 9 = Note On

Anmerkung: Note On mit Velocity=0 gilt als Note Off; davon wird häufig Gebrauch gemacht.

- Ah = Aftertouch
- Bh = Control Change
- Ch = Program (patch, instrument) change
- Dh = Channel Pressure
- Eh = Pitch Bend (Pitch Wheel)

Das höherwertige (linke) Nibble (dessen vorderstes Bit stets 1 ist) bietet mit der restlichen 3 Bits Platz für 8 Binärwerte (0 bis 7), in seiner Gesamtheit daher für 8 Binärwerte 8 bis Fh. Diese werden für die folgenden Befehle verwendet:

Anmerkung: Controller sind alle Einrichtungen, die eine gespielte Note beeinflussen, ausgenommen Note On und Note Off. Für das Modulation Wheel (Vibrato-Effekt) ist Controller Nr. 1 standardmäßig reserviert, alle anderen Controller-Nummern sind frei wählbar.

Das niederwertige (rechte) Nibble enthält die **Kanalnummer**. Diese kommt ausschließlich an dieser Stelle in den Statusbytes vor. Die nachfolgenden Datenbytes beziehen sich ausschließlich auf den geltenden Kanal.

Beispiel: 91 bedeutet: Note ein auf Kanal 1. Zur genaueren Bestimmung müssen darauf noch Datenbytes folgen, in diesem Beispiel zwei, nämlich für Tonhöhe (Pitch) und Lautstärke (Velocity).

Anmerkung: MIDI ist stark von der Vorstellung beeinflusst, dass MIDI-Daten auf einem Keyboard erzeugt werden. Die Schnelligkeit (velocity), mit der eine Taste niedergedrückt wird, ist maßgebend für die Lautstärke, daher der MIDI-Fachausdruck Velocity für Lautstärke.

Man sollte sich darüber im Klaren sein: Ein Akkord von zum Beispiel sechs Noten wird seriell gesendet und benötigt zur Übertragung so lange, dass zwischen der ersten und der letzten Note 3,2 ms vergehen (jede Note ist durch zwei Datenbytes gekennzeichnet, also 5 * 320 s Zeitunterschied, wobei nur ein einziges Statusbyte ("Note On") von 12 Datenbytes gefolgt ist. Bei rasch einsetzenden Tönen ("Attack" sehr plötzlich) kann das schon hörbar werden.

(1.2) Statusbytes der Kategorie Mode

Diese Statusbytes im Wertebereich B0h bis BFh werden auf einem implizit geltenden (mittels nicht standardisierten Verfahrens für ein Gerät festgelegten) Kanal empfangen, auf jedem anderen Kanal jedoch ignoriert. Mit ihnen kann im wesentlichen eingestellt werden, wie viele Stimmen (voices) auf welchen Kanälen der Empfänger akzeptiert. Auf das Statusbyte

- Note aus
- Note ein

gilt als Note Off; davon wird häufig

- Tastendruck nach dem Anschlag Controller einstellen
- Klangfarbe (Instrument) wählen
- durchschnittlicher Tastendruck
- Tonhöhenbeeinflussung, standardmäßig 2 Halbtöne

Tabelle 1: Höherwertiges Nibble des Statusbyte

folgen noch zwei Datenbytes. Näheres sollte dem Standard entnommen werden.

(1.3) Statusbytes allgemeiner Art

Statusbytes ab F0h (240d) leiten Messages ein, die sich nicht an einen bestimmten MIDI-Kanal richten und daher von jedem angeschlossenen Gerät "gehört" werden, sodaß jedes Gerät bei Bedarf reagieren kann. Im Gegensatz dazu werden die Messages der Kategorie Voice nur von einem Gerät gehört, das auf Empfangen auf dem bestimmten Kanal eingestellt ist, Messages auf einem anderen Kanal werden jedoch ignoriert. Also: nur der Empfänger, der auf diesen Kanal eingestellt ist, wertet die ankommenden Informationen auch aus. Alle anderen Informationen werden über die MIDI - Thru Buchse an das nächste Gerät weiter gegeben.

Diese Statusbytes können in zwei Kategorien eingeteilt werden:

(a) **System Common Messages**, F0h bis F7h, zum Beispiel Positionieren in einem Song, siehe **Tabelle 1** rechts oben.

Die "system exclusive" Message ist frei für individuelle Gerätetypen verwendbar.

(b) **System Real Time Messages**, F8h bis FFh, zum Beispiel Synchronisieren der MIDI-Geräte. Diese Messages dürfen an beliebigen Stellen mitten unter anderen, sogar mitten unter Datenbytes, liegen, haben jedoch selbst keine angehängten Datenbytes (sonst könnten sie ja ihrerseits wieder unterbrochen werden). Siehe **Tabelle 2** rechts oben.

Einige mögliche Statusbytes werden derzeit noch nicht verwendet und sollten von den MIDI-Geräten ignoriert werden.

(2) Erklärung einer elementaren MIDI-Datei

MIDI-Dateien werden überwiegend im standardisierten MIDI-Format (SMF - **Standard-MIDI-File**, .MID) abgespeichert: Es gibt davon die Typen 0, 1, 2; 1 ist der gebräuchlichste. Hintereinander stehen in der Regel mehrere Abschnitte oder Blöcke, in der Fachsprache Chunks heißen, nämlich Header-Chunks und Track-Chunks. Typ 1 enthält einen Header-Chunk und mehrere Track-Chunks (Spuren), deren jeder Messages (Meldungen) enthält, welche den musikalischen Ablauf beschreiben (Noten, Zeitinformationen, Klangfarbe, Spielanweisungen zu den Noten usw.). Ein Block (Chunk) muss die "Tempo Map" enthalten, mit der die Zeitdauer einer musikalischen Einheit (Viertelnote) festgelegt ist.

Alle MIDI-Dateien sollten Tempo und Takt ("Time Signature") enthalten; als Default würde ansonsten 120 Beats/min bzw. 4/4-Takt gelten. Da diese Angaben für das

angehängte Datenbytes		
F0h	system exclusive	Anzahl nach Bedarf
F1h	undefiniert	
F2h	song position	zwei, 14 bit, niederwertiges Byte zuerst
F3h	song select	eines, Song-Nummer
F4h	undefiniert	
F5h	undefiniert	
F6h	tune request	keines (für Analog-Synthesizer: Stimmen)
F7h	EOX (terminator)	keines (zum Beenden der system exclusive)

Tabelle 2: System Common Messages, F0 bis F7h

F8h	timing clock, 24 mal je Viertelnote zu senden	
F9h	undefiniert	
FAh	start	
FBh	continue	Zum Kontrollieren von Sequencern und Drum machines
FCh	stop	Zum Kontrollieren von Sequencern und Drum machines
FDh	undefiniert	
FEh	active sensing (praktisch nicht verwendet)	
FFh	system reset	

Tabelle 3: System Real Time Messages, F8h bis FFh

0000	4D 54 68 64 00 00 00 06 00 01 00 02 01 E0 4D 54	MThd.....MT
0010	72 6B 00 00 00 19 00 FF 51 03 06 1A 80 00 FF 58	rk.....Q.....X
0020	04 03 02 18 08 00 FF 59 02 02 00 00 FF 2F 00 4DY...../..M
0030	54 72 6B 00 00 00 5F 00 FF 21 01 00 00 FF 20 01	Trk..._..!.... .
0040	00 00 B0 07 7F 00 C0 28 00 90 3E 50 83 00 80 3E(>P...>
0050	40 60 90 42 50 83 00 80 42 40 60 90 45 50 83 00	@'.BP...B@'.EP..
0060	80 45 40 60 90 45 50 86 00 80 45 40 81 40 90 51	.E@'.EP...E@.@.Q
0070	50 83 00 80 51 40 60 90 51 50 86 00 80 51 40 81	P...Q@'.QP...Q@.
0080	40 90 4E 50 83 00 80 4E 40 60 90 4E 50 86 00 80	@.NP...N@'.NP...
0090	4E 40 00 FF 2F 00	N@../.

Tabelle 4: Dump einer Midi-Datei

Header: 0-3	MThd (Kennung)
4-7	Länge des eigentlichen Headers, immer 00 00 00 06
8-9	SMF, 00 01 für SMF-Typ 1 (hier beginnt der eigentliche Header)
A-B	Anzahl der Tracks
C-D	Time: Wenn Bit 15 = 0: Anzahl Delta-Time je Viertelnote 1: Time-Code-Verfahren
Beispiel Donauwalzer:	2 Tracks
0000	4D 54 68 64 00 00 00 06 00 01 00 02 01 E0
	M T h d Headerlänge SMF 1 1/260 Viertelnote je Delta-Time

Tabelle 5: Aufbau des Headers

ganze Musikstück gelten, werden sie als "Meta-Events" in die Datei geschrieben. In MIDI-Dateien des Typs 1 (den häufigsten) sollten sie im ersten Track stehen.

Auf den tatsächlichen Inhalt einer MIDI-Datei wird hier ohne Anspruch auf Vollständigkeit kurz eingegangen, da eine gewisse Vorstellung vom tatsächlichen Format der MIDI-Daten hilft, die Funktionsweise der MIDI-Software zu verstehen.

Tabelle 4 enthält eine MIDI-Datei der vier ersten Takte des Donauwalzers, einstimmig Violine (General MIDI) in D-Dur; Tem-

po 150 (Beats = Viertelnoten je Minute), das entspricht einer Dauer von 60/150 = 0,4 s je Viertelnote.

Vorbemerkung: Die erklärenden Abschnitte aus dieser MIDI-Datei sind vertikal genau mit dem nachfolgenden Speicherauszug ("Dump") ausgerichtet. Der Header ist kursiv, Track 1 unterstrichen, Track 2 normal gedruckt.

(a) **Aufbau des Headers** (siehe **Tabelle 5**):

24 MIDI-Clocks je Viertelnote sind Standard.

```

0000 4D 54 68 64 00 00 00 06 00 01 00 02 01 E0 4D 54 MThd.....MT
0010 72 6B 00 00 00 19 00 FF 51 03 06 1A 80 00 FF 58 rk.....Q.....X
0020 04 03 02 18 08 00 FF 59 02 02 00 00 FF 2F 00 4D .....Y...../..M
0030 54 72 6B 00 00 00 5F 00 FF 21 01 00 00 FF 20 01 Trk..._!.... .
0040 00 00 B0 07 7F 00 C0 28 00 90 3E 50 83 00 80 3E ...._(!..>P...>
0050 40 60 90 42 50 83 00 80 42 40 60 90 45 50 83 00 @'.BP...B@'.EP..
0060 80 45 40 60 90 45 50 86 00 80 45 40 81 40 90 51 .E@'.EP...E@.@.Q
0070 50 83 00 80 51 40 60 90 51 50 86 00 80 51 40 81 P...Q@'.QP...Q@.
0080 40 90 4E 50 83 00 80 4E 40 60 90 4E 50 86 00 80 @.NP...N@'.NP...
0090 4E 40 00 FF 2F 00                               N@!./..

```

Delta-Time = Anzahl Time-Ticks, die bis zur Ausführung des Events zu verstreichen haben, gemessen vom vorhergehenden Event. Höchstens 4 Bytes, das letzte hat 0 auf Bit 7, die vorhergehenden 1; Daten nur Bits 0 bis 6.

Beispiel Donauwalzer: Delta-Time ist für die Viertelnoten 8300h, für die halben Noten 8600h. Im vorderen Byte zählt das vorderste Bit nicht, der Zahlenwert ist daher 0300h bzw. 0600h, das entspricht 768 bzw. 1536 Delta-Time bis zum nächsten Event.

Eine Viertelnote erklingt 0,4 s lang; die Delta-Time geben ein relatives Maß für die Dauer anderer Noten, die halbe Note dauert demnach doppelt so lang.

Tabelle 4: Dump einer Midi-Datei

(b) Aufbau eines Tracks:

```

Track: 0-3 MTrk (Kennung)
      4-7 Länge des eigentlichen Tracks
      ab hier: Events
Beispiel Donauwalzer, erster Track:
0000                                4D 54
                                M T
0010 72 6B 00 00 00 19
      r k Tracklänge 25 Bytes
0010                                00 FF Kennung für Meta-Event
                                51 Kennung für Tempo
                                03 Länge (3 Bytes)
                                06 1A 80 = 400000 s je Viertelnote
0010                                00 FF Meta-Event
                                58 Time Signature
0020 04 (restliche) Länge des Events 03 Zähler im 3/4-Takt
      02 Nenner als Exponent zu 2, hier also 2 = 4
      18 Anzahl MIDI-Clocks je Metronom-Tick
      08 Anzahl 32stel-Noten je Viertelnote
      00 FF nächstes Meta-Event
      59 Key Signature (Tonart)
      02 (restliche) Länge des Events
      02 2 # (für D-Dur)
      00 Dur (01 wäre moll)
      00 FF nächstes Meta-Event
      2F Track-Ende
      00 immer 00

```

Hier endet der erste Track, der zweite schließt an:

```

0020                                4D                                M
0030 54 72 6B 00 00 00 5F
      T r k Tracklänge 5F = 95 Bytes
es folgen einige Meta-Events:
      00 FF 21 01 00                                diese beiden Meta-
      00 FF 20 01                                Events scheinen
      alle folgenden Meta-Events betreffen Kanal 1  ohne besondere
                                                    Funktion standard-
                                                    mäßig eingefügt zu
                                                    sein.

```

Es folgen jetzt Events der Kategorie Voice:

```

0040 00 00 B0 Controller Change auf Kanal 0
      07 Controller Nummer 7
      7F wird gesetzt auf den (standardmäßigen) Wert 7Fh
      00 C0 Program Change (Instrument wählen)
      28 General MIDI Instrument Nr. 40 (Violine)
      00 90 Note On auf Kanal 0
      3E Note Nr. 62 (D1)
      50 Velocity, durch das Einspielen
          bestimmt
      83 00 Delta-Time
      80 Note Off auf Kanal 0
      3E Note Nr. 62 (D1)
0050 40 Velocity, durch das Einspielen bestimmt. Die Taste wurde
      offensichtlich etwas langsamer losgelassen als angeschlagen
      60 Delta-Time (nur ein Byte, weil sehr kurze Zeit)
      90 Note On auf Kanal 0
      42 Note Nr. 66 (Fis1)
      50 Velocity, durch das Einspielen bestimmt
      83 00 Delta-Time
      80 Note Off auf Kanal 0
      42 Note Nr. 66 (Fis1)
      40 Velocity
      60 Delta-Time usw.

```

➤➤ *Schluss "Audioschnitt"*

- Einspuriges Stereo-VIP für das herzustellende Gesamt-File öffnen.
- VIP laden, gewünschten Bereich (der übernommen werden soll, also im Normalfall so gut wie alles) markieren.
- Mit Werkzeuge - Track Bouncing (nur markierter Bereich) die .WAV-Datei herstellen.
- Gesamtes entstandenes .WAV-File markieren und aus dem HDP-Fenster in das Gesamt-VIP kopieren (mittels Drag and Drop).
- Die nicht mehr benötigten Fenster (HDP und VIP) schließen.
- Weitere .WAV-Files auf die gleiche Weise laden.
- Eventuell Pausenzeit auf einen anderen Wert als die standardmäßigen 2 Sekunden einstellen: Werkzeuge - CD - Pause-Zeit einstellen.
- Werkzeuge - CD - Indizes an Objektkanten setzen. Damit werden die Indizes (Track Marker) auf die Titellänge gesetzt.
- Werkzeuge - CD - Erstelle CD. Damit wird das TOC erzeugt. Dabei gibt es drei Varianten:
 - (1) *keine Berechnungen*: Fades, Crossfades, Lautstärkekurven u.dgl. werden nicht berechnet. Das Verfahren ist daher nur anwendbar, wenn von diesen virtuellen Funktionen nicht Gebrauch gemacht wurde.
 - (2) *nutze Original-Dateien*: Dieses Verfahren ist zu empfehlen, wenn im VIP von virtuellen Funktionen Gebrauch gemacht wurde. Die errechneten Änderungen werden in den Original-Dateien abgespeichert; es empfiehlt sich daher, von diesen vorher Sicherungskopien anzulegen.
 - (3) *komplett neue Datei*: Track-Bouncing wird automatisch ausgeführt, eine neue .WAV-Datei wird erzeugt. Dieses Verfahren ist anzuwenden, wenn die Vorbereitung der Daten für die anderen Verfahren noch nicht ausreicht (Amplitude macht eine diesbezügliche Meldung). Allenfalls muss man noch alles im VIP-Fenster markieren (A-Taste) und Werkzeuge - CD - Bouncing (interner Mixdown) anwenden.

Werkzeuge - CD - TOC anzeigen. Das TOC kann hier überprüft und eventuell (mit größter Vorsicht!) auch editiert werden.