

# Musical Instrument Digital Interface

---

**Musical Instrument Digital Interface**, [ˈmjuzɪkəl<sup>ⓘ</sup> ˈɪnstɹəmənt<sup>ⓘ</sup> ˈdɪdʒɪtəl<sup>ⓘ</sup> ˈɪntəˌfeɪs<sup>ⓘ</sup>], (engl. „digitale Schnittstelle für Musikinstrumente“), kurz: **MIDI** [ˈmiːdi<sup>ⓘ</sup>] ist ein Industriestandard für den Austausch musikalischer Steuerinformationen zwischen elektronischen Instrumenten,<sup>[1]</sup> wie z. B. Keyboards oder Synthesizern. Dieser Standard umfasst sowohl die genaue Beschaffenheit der erforderlichen Hardware als auch das Kommunikationsprotokoll für die zu übermittelnden Daten.<sup>[2]</sup> MIDI 1.0 wurde im August 1982 eingeführt<sup>[3]</sup> und ist inzwischen mehrfach erweitert worden. Entwickelt wurde MIDI von Dave Smith in Kooperation mit Ikutaro Kakehashi von der Roland Corporation, wofür beide im Jahr 2012 mit dem technischen Grammy ausgezeichnet wurden.<sup>[4]</sup>

## Inhaltsverzeichnis

---

- 1 **Einsatzbereiche**
- 2 **Funktionsweise**
- 3 **Technik**
  - 3.1 Anschlüsse
  - 3.2 Eingabegeräte
  - 3.3 Rechner als Speichermedien
    - 3.3.1 Signal-Konverter
    - 3.3.2 Commodore 64
    - 3.3.3 Atari ST
    - 3.3.4 Commodore Amiga
    - 3.3.5 IBM-PC
    - 3.3.6 Allgemeines
  - 3.4 Expander (Soundmodul)
    - 3.4.1 Samplebasierte Klangerzeugermodule
      - 3.4.1.1 ROM-Klangerzeuger (*ROMpler*) und Sampleplayer
      - 3.4.1.2 Sampler
      - 3.4.1.3 Samplebasierte Synthesizermodule
    - 3.4.2 Synthesizermodule mit generischer Klangerzeugung
    - 3.4.3 Steuermöglichkeiten der Expander
    - 3.4.4 Bauweise
  - 3.5 Sequenzer
  - 3.6 Zusätzliche Hardware
- 4 **Protokoll**
  - 4.1 Nachrichtentypen
  - 4.2 Controller
  - 4.3 Herstellerspezifische Erweiterungen
  - 4.4 Systemexklusive Meldungen
- 5 **Synchronisation**
  - 5.1 1. Synchronisation von Systemen
  - 5.2 2. Synchronisation von Synthesizerfunktionen und Effektgeräten

5.3	MIDI-Clock und Song Position Pointer
5.4	MIDI-Time-Code
<b>6</b>	<b>Dateiformate</b>
<b>7</b>	<b>Alternative Übertragungswege</b>
7.1	USB und FireWire
7.2	Wireless MIDI
7.3	IP-basierende Netzwerke
7.4	Andere
<b>8</b>	<b>Kritik</b>
8.1	Vor- und Nachteile gegenüber digitaler Audioaufzeichnung
8.2	Latenz
8.3	Stimmungen und Skalen
8.4	Auflösung
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Weblinks</b>
<b>11</b>	<b>Quellen</b>

# Einsatzbereiche

---

Das MIDI-Protokoll wurde ursprünglich zur Kommunikation von Synthesizern unterschiedlicher Hersteller entwickelt. Der eigentliche Sinn war, von einer Tastatur eines Synthesizers aus weitere Synthesizer anzusteuern. Davor konnten Synthesizer nur analog und mit großem Verkabelungsaufwand verbunden werden.

Zu der damaligen Zeit hatten die Synthesizer nur wenige Stimmen, d. h., sie konnten meist nur 4–8 Töne gleichzeitig erzeugen. Trotz einer gewissen Soundauswahl konnte kein Gerät mehr als einen Sound gleichzeitig erzeugen. Wollte man also zwei oder mehrere Sounds mit einem Tastendruck spielen, musste man zwei Geräte mit einer Tastatur verkoppeln. So konnte man unterschiedliche Sounds übereinanderlegen, um z. B. einen „dickeren“ Synthesizerstreicherklang zu bekommen oder Synthesizerstreicher mit Synthesizerbläsern zu kombinieren.

Das war nun mit der Verbindung über ein einzelnes MIDI-Kabel möglich, indem der MIDI-Out des Hauptgerätes mit dem MIDI-In des angesteuerten Gerätes per 5-poligem MIDI-Kabel verbunden wurde (wobei nur zwei Pole genutzt werden). Da die Audiosignale der verschiedenen Synthesizer keine MIDI-Steuerdaten sind, müssen diese über zusätzliche Leitungen einem Mischpult zugeführt werden.

MIDI trennte auch gleichzeitig die Tastatur eines Synthesizers von seiner Klangerzeugung, was natürlich die Einsatzmöglichkeiten eines Instrumentes massiv erhöhte: Denn so war es auch möglich, eine Tastatur aufzuteilen (splitten) und die Tastaturbereiche auf verschiedene Synthesizer zu verteilen. So konnte der Keyboarder z. B. mit dem linken Tastaturbereich einen Streicherklang mit einem angesteuerten Synthesizer und mit der rechten Hand einen Solo-Synthesizerklang mit dem lokalen Gerät spielen.

Schnell wurde die MIDI-Schnittstelle für fast jede Art an elektronischen Musikinstrumenten adaptiert, so z. B. für Expandermodule, Sampler, Drumcomputer, Effektgeräte (Hall, Echo, Equalizer usw.), Hardwaresequencer (Aufnahme- und Abspielgeräte für MIDI-Daten), Computer, *Controller* (wie Masterkeyboards, Drum-Pads, Masterkeyboardcontroller, Standard-Midi-File-Player, Fader-Boxen, später auch für Sound- und Audiokarten usw.), nicht zuletzt auch – zweckentfremdet – zur Steuerung von Lichteffekten für Bühnen (MIDI Show Control).

Der Einsatz von Computern in der Tonstudioteknik gab MIDI einen weiteren Schub. So konnte der wenig versierte Keyboarder mit Hilfe eines Hardwaresequencers bzw. des Computers und eines Sequencerprogrammes komplexe, schwierige oder gar manuell unspielbare Musikstücke erstellen, weil er die MIDI-Daten im Sequencer verändern und korrigieren konnte. Dadurch, dass nur die Steuerdaten gespeichert werden, kann der Sound auch nach einer Aufnahme im Sequencer beliebig ausgetauscht werden. Das ergab völlig neue Möglichkeiten – auch für versierte Musiker – und hat Auswirkungen auf die Produktionsweise von Musik bis heute:

Komposition, Arrangement und Notensatz wurden durch die Verbindung von MIDI-fähigem Keyboard und Computer erheblich vereinfacht. Variationen von Stimmen und Songabläufen sind sehr schnell realisierbar und bleiben jederzeit änderbar. Diese Zeitersparnis ist u. a. bei Studioproduktionen ein wichtiger Faktor. Der Komponist greift oft zwar auf das Hilfsmittel Computer zurück und editiert sein Konzept direkt über Software, viele Stimmen werden jedoch nach wie vor über eine Klaviertastatur bzw. ein Masterkeyboard eingespielt.

Mit speziellen Wandler-Geräten kann man auch aus den Tönen beliebiger akustischer Instrumente wie Gitarre oder Saxophon MIDI-Daten erzeugen. Dabei muss aus einem komplexen Klangmuster die gespielte Tonhöhe ermittelt werden, was, abhängig vom Instrument und der Spielweise, bald an Grenzen stößt. Bei einer Gitarre z. B. muss interpretiert werden, ob ein Ton durch einen Bundwechsel oder Ziehen einer Saite entstanden ist. Aber für viele Anwendungen ist das nicht erforderlich, und so kann man mit einem akustischen Instrument und einem über MIDI angeschlossenen Synthesizer oder Sampler völlig andere Sounds in Kombination oder eigenständig erzeugen.

In den 2000er Jahren, als der Speicher in Mobiltelefonen noch knapp war, benutzte man das MIDI-Format auch für Klingeltöne.

## Funktionsweise

---

Spielt man auf einem Keyboard eine Taste, werden digitale Informationen über Tonhöhe und Anschlagstärke am MIDI-Ausgang des Keyboards ausgegeben und lassen sich an den MIDI-Eingang eines Computers übermitteln oder zur Steuerung der Klangerzeuger in elektronischen Instrumenten und Soundkarten verwenden. Solche Befehle sind beispielsweise *Note-on* („Taste für Note x wurde gedrückt“) und *Note-off* („Taste für Note x wurde wieder losgelassen“).

Ein vollständiger 3-Byte-Datensatz für einen Spielbefehl für eine Note könnte beispielsweise wie folgt aussehen:

1. Byte: *Note ein* auf MIDI-Kanal 1
2. Byte: Note *C3*
3. Byte: Anschlagstärke 103

Diesen Ton spielt ein angesteuerter Klangerzeuger so lange, bis dieser den 3-Byte-Befehl mit einem *Note aus*-Byte anstelle des *Note ein*-Byte empfängt. Welchen Sound der Klangerzeuger spielt, wird entweder zuvor am Klangerzeuger oder mit weiteren MIDI-Befehlen vor dem Spielbefehl für die Note eingestellt.

Ein Computer kann diese Informationen aufzeichnen, abspeichern und in verschiedenen Editoren visualisieren, eingeben und manipulieren. Üblich sind hierbei ein Listeneditor, in dem MIDI-Daten an sich editiert werden, der wohl am meisten verwendete Piano-Roll-Editor, der eine Klaviertastatur mit einem Zeitverlauf darstellen kann und ein Noteneditor, der die Notenschrift auch auf dem Bildschirm sichtbar macht. Letzterer ist aber oft in der Praxis nicht zu gebrauchen, weil das Notenbild völlig zerrissen aussieht, wenn Noten manuell ohne weitere Bearbeitung eingespielt werden.

Gleichzeitig oder auch später können die aufgezeichneten Daten an ein MIDI-Instrument zurückgesendet werden. So wird die Eingabe von Spielinformationen von der Tonerzeugung getrennt.

Neben den musikalischen Befehlen können weitere Datenpakete zur Steuerung des Zielgerätes genutzt werden, so etwa *Program-Change-Befehle* zur Auswahl eines seiner meist vielen hundert Klangspektren. Viele Klangerzeuger wie Synthesizer, Expander und andere verstehen Befehle, mit denen ihre interne Klangerzeugung direkt beeinflusst werden kann, um so aus einer Reihe einfacher Grundschwingungsformen komplexe, individuelle Klänge zu erzeugen.

Inzwischen nutzen neben elektronischen Instrumenten auch zahlreiche andere Geräte wie beispielsweise (digitale) Mischpulte und Mehrspuraufzeichnungsgeräte das MIDI-Protokoll, um darüber Steuerinformationen auszutauschen. Anstatt MIDI-Datenpakete zur Übertragung von Notenbefehlen zu nutzen, können die Daten hier beispielsweise zum Fernbedienen sämtlicher Mischpultfunktionen (Fader, Muteschalter, Panorama, etc.) oder zur Steuerung der Laufwerksfunktionen eines Recorders (Play, Stopp, Vor-/Rückspulen) verwendet werden.

# Technik

MIDI verwendet ein unidirektionales Protokoll zur seriellen Datenübertragung ohne Datenflusskontrolle. Die Übertragungsgeschwindigkeit beträgt 31250 Bit/s (exakt 32 µs pro Bit). Zu jedem Byte, bestehend aus 8 Bit, kommt ein Start- sowie ein Stopp-Bit, sodass die komplette Übertragung eines Datensatzes bestehend aus 30 Bits 960 µs dauert. Bis auf das Fehlen des Parity-Bits entspricht es damit dem Protokoll bei PC-UARTs.

Im Unterschied zu pegelgesteuerten Schnittstellen wird bei MIDI jedoch eine 5-mA-Stromschleife verwendet. Physisch sind die Anschlüsse als fünfpolige DIN-Buchsen (DIN 5/180° – früher Diodenbuchse/-stecker) realisiert. Die Pins 4 und 5 werden als Datenleitung benutzt. Dabei liegt gemäß der MIDI-Spezifikation Pin 4 über einen 220-Ohm-Widerstand an +5 V. Pin 5 kann über 220 Ohm mit 0 V verbunden werden. Ist am anderen Ende ein MIDI-In-Port angeschlossen (nochmals 220 Ohm in Reihe mit einem Optokoppler), fließt ein Strom über die Leitung, dieser Zustand ist als logische „0“ definiert. Wird Pin 5 nicht verbunden, fließt kein Strom, was als logische „1“ gilt und auch den Ruhezustand darstellt.

Laut einer nachträglichen Erweiterung<sup>[5]</sup> der MIDI-Spezifikation sind nun auch MIDI-Ausgänge möglich, die auf einer 3,3 V-Spannungsversorgung beruhen. In diesem Fall werden Serienwiderstände von 33 Ohm und 10 Ohm an Pin 4 bzw. an Pin 5 eingesetzt.

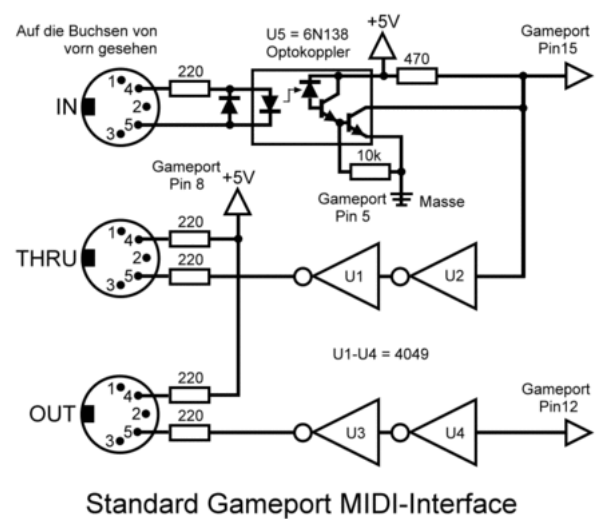
Durch den Optokoppler in der Empfangsleitung ergibt sich eine galvanische Trennung der einzelnen MIDI-Geräte untereinander, die Masseleitung (Pin 2 und Kabelabschirmung) darf an der MIDI-In-Schnittstelle nicht angeschlossen werden, um Masseschleifen zu vermeiden.

In manchen Ausnahmen (z. B. bei der MIDI-Interface-Karte „Roland MPU-401 AT“ als ISA-Karte) sind die Anschlüsse auch als 6-polige Mini-DIN-Buchsen ausgelegt. In solchen Fällen hilft ein Anschlussadapter, der baugleich zu einem Tastaturadapter „Mini-DIN-Stecker zu DIN-Buchse“ (PS/2 auf AT) ist.

## Anschlüsse

Es existieren drei verschiedene Arten von MIDI-Anschlüssen, *MIDI-In*, *MIDI-Out* und *MIDI-Thru*.

- *MIDI-In* wird von einem Gerät zum Empfang verwendet.
- *MIDI-Out* wird zum Senden verwendet.

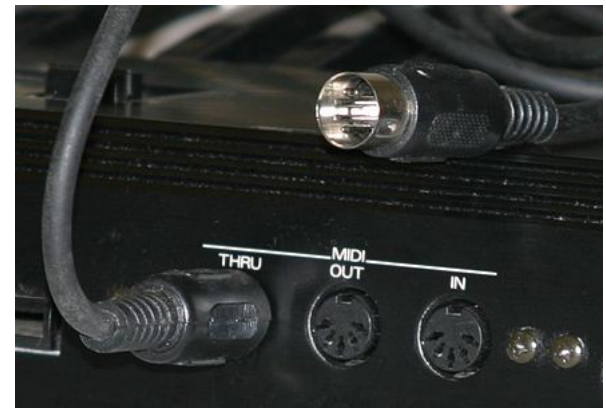


MIDI-Interface am PC: externer Teil zum Anschluss an die 15-polige kombinierte MIDI-/Game-Buchse

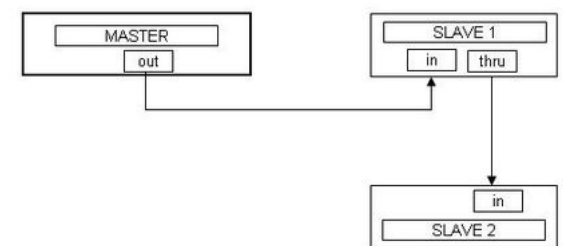
- *MIDI-Thru* schickt am MIDI-In empfangene Signale unbearbeitet weiter.

MIDI arbeitet nach dem Master-Slave-Prinzip. Will man mit einem Keyboard einen Synthesizer steuern, verbindet man die MIDI-Out-Buchse des Keyboards (Master) mit der MIDI-In-Buchse des Synthesizers (Slave). Sollen mit einem Keyboard (Master) zwei Soundmodule als Slave A und B angesteuert werden, verbindet man die MIDI-Out-Buchse des Masters mit der MIDI-In-Buchse des Slave A sowie die MIDI-Thru-Buchse des Slave A mit der MIDI-In-Buchse des Slave B.

Ein häufig anzutreffendes Szenario ist der Einsatz eines Computers mit entsprechender Sequenzer-Software sowie der Anschluss eines Keyboards oder elektronischen Pianos zum Einspielen der Noten und mehreren Synthesizern zur Klangerzeugung. Dabei wird üblicherweise die MIDI-Out-Buchse des Keyboards mit der MIDI-In-Buchse des Computers verbunden, die MIDI-Out-Buchse des Computers mit den MIDI-In-Buchsen der Synthesizer, ggf. verkettet über die MIDI-Thru-Buchsen. Zu beachten ist, dass sich dabei die unvermeidlichen Verzögerungen im MIDI-Datenstrom summieren und damit zu Timingfehlern führen können. Eine sternförmige MIDI-Verkabelung, bei der das Master-Keyboard seine Daten an einen zentralen Verteiler (MIDI-Patchbay) sendet, an dem alle weiteren MIDI-Geräte angeschlossen sind, beseitigt solche Probleme.



MIDI-Anschlüsse



Master-Slave-Prinzip

## Eingabegeräte

Ein Masterkeyboard erzeugt Noteninformationen im MIDI-Format und dient ausschließlich der Steuerung von Expandern, Software-Synthesizern oder zur Aufzeichnung der Tastenbewegung beim Einspielen von Musik in Sequenzer. Es enthält keine eigene Klangerzeugung. Die Klangsteuerung an den Geräten kann über gerätespezifische Steuerfunktionen wie Bankumschaltung geschehen.



Masterkeyboard mit 88 gewichteten Hammermechanik-Tasten

Dem gegenüber stehen die reinen MIDI-Controller. Dabei handelt es sich um Geräte ohne Tastatur, die lediglich Knöpfe und Drehregler besitzen, mit denen eingehende Daten modifiziert oder dem Datenstrom neue Daten auf anderen Kanälen hinzugefügt werden. Sie werden zwischen ein Masterkeyboard und einen Empfänger – oder parallel dazu geschaltet.

Heute werden oft kombinierte Eingabegeräte verwendet, die sowohl Noten- als auch umfangreiche Kontrollfunktionen ausüben können. Einige erzeugen reine MIDI-Informationen, andere sind zusätzlich oder alleinig an den PC anschließbar.

Für viele akustische Musikinstrumente existieren Tonaufnehmer zur Erzeugung von MIDI-Signalen (z. B. Guitar-to-MIDI-Converter, Piano-Aufsetzer etc.) Hier wird die akustische Schwingung durch ein Mikrofon aufgenommen und in eine MIDI-Tonhöhe umgerechnet, indem ein Grundton ermittelt und Controllerwerte zur Modulation desselben



MIDI-Controller für die Nutzung am PC



erzeugt werden. Damit sind Töne variierender Tonhöhe (Vibrato) erzeugbar. Einige dieser Systeme eignen sich auch zum Aufnehmen des menschlichen Gesangs. So können komplexe Stimmverläufe wie blue notes und Phrasierungen z. B. durch Pfeifen auf das Notenpapier gebracht werden.

Eine Reihe von Instrumenten existieren heute als reine MIDI-Geräte, bei denen die Tonerzeugung ausschließlich mit einem Expander möglich ist. Beispiele dafür sind MIDI-Geigen, MIDI-Gitarren, Blaswandler und das MIDI-Schlagzeug.

## Rechner als Speichermedien

### Signal-Konverter

Um mit einem Computer über MIDI zu kommunizieren, muss ein *Signal-Konverter* zwischengeschaltet werden, der gewöhnlich als MIDI-Interface bezeichnet wird. Er übersetzt die Spannungspegel und sorgt für eine galvanische Entkopplung. Im Prinzip kann jede serielle Datenübertragungs-Schnittstelle eines Computers mit einem geeigneten MIDI-Interface für die MIDI-Übertragung genutzt werden.

### Commodore 64

Eine Pionierrolle spielte der Commodore 64, auf dem insbesondere die deutschen Softwareautoren Gerhard Lengeling und Karl Steinberg ihre ersten Sequencer programmierten, die für die Namen C-LAB, Emagic und Steinberg stehen.

### Atari ST

Der kommerzielle Durchbruch für MIDI als Plattform für professionelle Musikproduktion ist eng mit dem Atari ST verbunden, da dieser standardmäßig mit einer MIDI-Schnittstelle ausgeliefert wurde. Die Entwicklung wichtiger MIDI-Programme wie Cubase (Steinberg) oder Notator (Lengeling) begann auf dem Atari ST.

### Commodore Amiga

Auf dem Commodore Amiga prägte die Softwarefirma Blue Ribbon Inc. mit *Bars & Pipes Professional* ein neues Sequencer-Software-Prinzip, das durch seine frei programmierbare Plugin-Schnittstelle in seinen Funktionen fast beliebig erweiterbar ist. Die meisten MIDI-Interfaces für den Commodore Amiga wurden als Adapter für die serielle Schnittstelle angeboten und sind mit einem MIDI-In, einem MIDI-Thru und meistens drei MIDI-Out ausgestattet. Es gibt sowohl synchrone als auch asynchrone MIDI-Interfaces. Bei einem asynchronen MIDI-Interface sind die verschiedenen MIDI-Out-Schnittstellen unabhängig voneinander ansteuerbar. Bei drei MIDI-Out-Schnittstellen gibt es also 48 MIDI-Kanäle ( $3 \times 16$ ).

### IBM-PC

Hier handelte es sich ursprünglich um eine 8-Bit-ISA-Steckkarte des Herstellers Roland. Viele für MS-DOS-PCs erhältliche Computerspiele zwischen 1988 und 1995 unterstützen diese MIDI-Schnittstelle zur Ansteuerung von Klangerzeugern wie z. B. der internen Roland LAPC-I oder dem externen MT-32. Andere Hersteller wie bspw.



Starr Labs Ztar Z6 series (drvinay's studio)

Creative Labs unterstützten den MPU-401-Modus nur eingeschränkt im so genannten Dumb-Mode (*UART*), während der *Intelligent-Mode*, der genaues Timing durch Hardwareunterstützung garantierte, nur von Rolands eigenen Produkten beherrscht wurde.

Die standardmäßigen DIN-Buchsen für MIDI sind zu groß, um direkt in die Rückplatte einer PC-Steckkarte eingebaut zu werden. Lange Zeit war die übliche Vorgehensweise, die MIDI-Signale, die an einem kombinierten Game-/MIDI-Anschluss entsprechender Soundkarten verfügbar waren, über einen Adapter (siehe MIDI-Anschlüsse) auf die Standard-MIDI-Schnittstelle umzusetzen. Ältere PC-Soundkarten, ausgehend vom Soundblaster, haben einen Anschluss geprägt, bei dem sich Game-Interface und MIDI-Interface eine 15-polige D-Sub-Buchse teilen und der heute immer noch in billigeren, nicht professionellen MIDI-Interfaces in PCs vertreten ist. Die Soundkarte braucht dabei nur zwei digitale, serielle Leitungen ohne Datenflusskontrolle zur Verfügung zu stellen (MIDI verwendet keine Datenflusskontrolle). Bei dieser Art der Hardware-Implementierung ist ein Teil des MIDI-Interfaces in einen externen, oft separat zu erwerbenden Teil verlegt, der meist in dem dickeren Stecker eines Kabels vergossen ausgeführt ist. Motherboards, die Sound-, MIDI- und Game-Controller on-Board haben, haben diese kombinierte Game-/MIDI-Anschlussbuchse übernommen. Dem entsprechen Sound-, Game- und MIDI-Chipsätze, die diese Funktionalitäten teilweise oder ganz gemeinsam integrieren. Das Vorhandensein einer 15-poligen D-Sub-Buchse an sich erlaubt jedoch noch keinen Rückschluss darauf, ob ein MIDI-Interface vorhanden ist oder, falls vorhanden, von welcher Qualität es ist.

### Allgemeines

Softwareseitig war die Hardware meistens MPU-401-kompatibel. Vorher waren auch MIDI-Interfaces für die serielle (COM) und parallele (Druckerport) Schnittstelle im Gebrauch. Professionelle MIDI-Geräte für PCs benutzen oft proprietäre (herstellerspezifische) Buchsen zwischen Steckkarte und dem externen Gerät. Inzwischen gibt es jedoch viele MIDI-Interface-Geräte für USB, FireWire (mLAN) und LAN.

### Expander (Soundmodul)

Ein Expander ist ein externer Klangerzeuger ohne eigene Tastatur. Er empfängt Noten ausschließlich per MIDI. Nur Parameter, die sich mit Knöpfen am Expander einstellen lassen, werden zurück übertragen und können aufgezeichnet werden.



Synthesizer

Ein Expandermodul erweitert die Möglichkeiten eines Tastatur-Synthesizers oder Keyboards. Wie ihre Pendants mit Tastatur können Expander viele Formen der Klangerzeugung zur Verfügung stellen. Wegen ihres geringen Platzbedarfs sind sie von ihrer Bedeutung her mindestens ebenso relevant wie die Tastaturversionen für die Entwicklung und Verbreitung von MIDI.



Synthesizer als Expander

Die Expandertechnik bietet die Möglichkeit, in relativ kleinen Geräten eine Vielfalt von Klängen zu erzeugen. Ohne sie hätte sich Midi sicher weniger schnell und weniger weit verbreitet, denn andere Verfahren verbieten sich vielerorts wegen ihres Platzbedarfs und ihrer Kosten.

Die Klangerzeugungsverfahren von Expandermodulen kann man in zwei Grundklassen aufteilen:

1. samplebasierte Klangerzeugermodule;
2. Synthesizermodule

## Samplebasierte Klangerzeugermodule

### ROM-Klangerzeuger (*ROMpler*) und Sampleplayer

Diese Klangerzeuger stellen eine Vielzahl unterschiedlicher Grundklänge (Multisamples) zur Verfügung, die einfach abgespielt werden. Sie dienen dazu, einen Grundvorrat an Imitationsklängen natürlicher Instrumente bereitzustellen. Deshalb waren diese Geräte von Anfang an mit mindestens 16 Stimmen und mit mindestens 6 Klangfarben (*Timbres*) ausgestattet. Dank dieser *Multitimbralität* können mehrere unterschiedliche Klänge auf unterschiedlichen Midikanälen abgerufen werden. In der Anfangszeit war dies nur mit wenigen Geräten möglich.

Diese Klangerzeugerklasse war zum Beginn der Expanderzeit am weitesten verbreitet. Beispiele sind *Roland U110* und *Roland U220*. Etwas später bekamen diese Geräte einfache Bearbeitungsfunktionen, also einfache Filter. Dadurch wurden sie jedoch nicht zu ausgewachsenen Synthesizern, denn diese Filter dienten mehr zur Klangverfeinerung als zur echten Neuschaffung von Klängen. Ein Beispiel ist der *Korg M1r*.

### Sampler

Sampler stellen keine Klänge zur Verfügung, sondern können diese aufnehmen und abspielen. Meist werden Samples aus einer Sound-Library in das Gerät geladen. Mit Samplern kann der Benutzer eigene Multisamples erstellen und so eine aufwändig zusammengestellte Instrumentenimitation schaffen. Sampler können auch Bearbeitungsfunktionen wie Loops bereitstellen. Sampler lassen neuartige Spielweisen für Musiker zu, beispielsweise das Verwenden von Drum-Loops und One-Shot-Effektsounds innerhalb einer Keymap, das heißt einer Tastaturzusammenstellung von verschiedenen Samples.

Die Samplertechnik hat die moderne Musik maßgeblich geprägt. Musikstile wie Hip Hop wären ohne Sampler nicht vorstellbar. Einen großen Einfluss hat diese Technik auch auf die Filmmusik. Gute Orchesterimitationen sind bis heute nur mit aufwändiger Samplingtechnik möglich.

Sampler wurden und werden auch im Live-Betrieb oft verwendet, beispielsweise um die Klangeigenschaften eines echten Schlagzeugs durch Samples zu erweitern, oder um Chöre oder Backing-Vocals bei kleinen Bandbesetzungen bereitzustellen. Die Chorsamples drückt ein Keyboarder mit einer Taste ab, oder der Schlagzeuger löst sie mit einem E-Pad per MIDI-Befehl aus.

Fast alle weitverbreiteten Sampler wurden nur als Expanderversionen ausgeführt. Ein bekanntes Beispiel ist die Akai-S-Serie.

### Samplebasierte Synthesizermodule

Es gibt eine Vielzahl von samplebasierten Synthesetechniken:

- Die *LA-Synthese* ist ein von der Firma Roland entwickeltes Syntheseverfahren. Sie trennt Klänge auf in Transienten und Flächenklänge, die dann mit einem digitalen Filter bearbeitet werden. Über die verschiedenen Zusammenstellungen dieser Klänge können neue Klänge erzeugt werden. Ab dem D50 und seinem Expandermodell D550 basieren viele Modelle der Firma bis heute auf diesem Klangerzeugungsverfahren.
- Die Firma Yamaha hingegen integrierte Samples in ihr FM-Verfahren, in dem ROM- und RAM-Samples Wellenformen bereitstellen.
- Korg hatte mit der *Wavestation* ein Gerät, das Samples rhythmisch abspielen konnte.
- Bei Geräten der Firmen Waldorf und Ensoniq gab es das Wavetable-Verfahren, bei dem einzelne Zyklen von Samples durchlaufen werden und so neue Klänge erzeugt werden.
- Kurzweil entwickelte mit der VAST-Synthese ein Verfahren, mit dem Samples über aufwändige DSP-Funktionen extrem ummoduliert werden und damit verfremdet werden, sodass das Ausgangssample nicht mehr zu erkennen ist.



Jede Klangerzeugung hat einen eigenen Charakter. Eine Auswahl an Geräten mit Tastatur live oder im Tonstudio bereitzustellen, ist aus Platzgründen oft nicht möglich, weshalb hier meist Expander zum Einsatz kommen.

## Synthesizermodule mit generischer Klangerzeugung

Diese Geräteklasse ist ähnlich umfassend, wie die der samplebasierten Synthesizer, wobei die Klangerzeugung rein *synthetisch* geschieht. Sie reicht von modularen, analogen Schrankwänden, beispielsweise von Doepfer, bis hin zu digitalen Kleinstmodulen in Halbzollgröße.

Viel verwendete Syntheseverfahren sind (kleine Auswahl):

- analoge Synthese mit analogen Oszillatoren und Filtern (Roland MKS 80)
- analoge Synthese mit digital kontrollierten Oszillatoren und analogen Filtern (Oberheim Matrix6)
- FM-Synthese (Yamaha DX7/TX802)
- additiven Synthese (Kawai K5000R)
- virtuellen Syntheseverfahren aller Art:

Virtuelle Simulationen sind die modernste Art der Synthese, sie bilden akustische, mechanische, elektrische und elektronische Bauweisen der Instrumente nach, weshalb sich ihre Bedienung an den baulichen Eigenheiten orientiert und deshalb völlig verschieden ist von den klassischen Syntheseverfahren. Ein Beispiel wäre das virtuelle Öffnen des Flügeldeckels, was einen mechanischen Vorgang widerspiegelt. Für einen Filter hingegen gibt es keine echten Gegenpart, er bleibt ein elektronischer Vorgang, der von der Charakteristik der elektronischen Schaltung abhängt.

Beispiele für virtuelle Synthese sind „virtuell-analoge“ Synthesizer, „virtual Strings“ für Gitarren-, Bassgitarren- und Streichinstrument-Simulationen; „virtual Brass“ für Blasinstrument-Simulationen, diverse Orgel-, Piano- und E-Piano-Simulationen für mechanische und elektromechanische Instrumente.

## Steuerungsmöglichkeiten der Expander

Unterschiedliche Synthesetechniken erfordern unterschiedliche Bearbeitungsweisen. Mittlerweile sind viele Expandermodule mit Steuerelementen wie Reglern, Fadern und Schaltern ausgestattet, die MIDI-Signale senden können. So können beispielsweise Regelverläufe eines Filters vom Regler eines Expandermoduls an einen Sequenzer zur Aufzeichnung gesendet werden. Ihre Midi-Noten-Befehle können sie durch den Sequenzer oder von einer Tastatur erhalten. Viele dieser Geräte sind daher je nach Situation Sender, Empfänger oder beides gleichzeitig.

## Bauweise

Expandermodule werden normalerweise in genormter 19-Zoll-Bauweise hergestellt. Es gab und gibt immer wieder Ausnahmen. In den Anfangszeiten von MIDI gab es noch häufiger Geräte, die nicht dieser Norm entsprachen.

## Sequenzer

Der Hardware-Sequenzer dient der Aufzeichnung der MIDI-Daten und dem Arrangement eines Musikstückes. MIDI-Sequenzer erlauben das Programmieren, die Aufzeichnung sowie die Wiedergabe von aufgezeichneten oder programmierten MIDI-Informationen (Notenwerte, Anschlagsstärke sowie weiteren Steuerungsbefehlen wie z. B. Modulation). Für den Live-Einsatz erfreuen sich auch die in Keyboards oder Groove-Boxes integrierten Sequenzer großer Beliebtheit. Eine Kombination aus Klangerzeuger und Synthesizer, Masterkeyboard und Hardware-Sequenzer wird als Workstation bezeichnet.

*Softwaresequenzer* haben im Bereich der Komposition große Bedeutung, da sie über die Standardfunktionen (Programmieren, Aufzeichnen, Abspielen) hinaus auch weitere Bearbeitungsmöglichkeiten in grafischer Form bieten (nachträgliches Editieren, Quantisierung usw.) und heutzutage nicht nur MIDI-, sondern auch Audiomaterial verarbeiten können. Diese Kombination aus Audio- und MIDI-Bearbeitung auf einem PC nennt man DAW (Digital Audio Workstation).

Die heute überwiegend verwendeten Sequenzerprogramme sind das bereits erwähnte Cubase von Steinberg für Mac (OSX) und Windows-PC und sein Pendant Logic, das, inzwischen von Apple aufgekauft, nur noch für Apple Macintosh bezogen werden kann. Steinberg (einschließlich Cubase) wurde inzwischen von Yamaha aufgekauft. Daneben gibt es Rosegarden und MusE auf unixartigen Plattformen und einige weitere Lösungen wie Sonar, Ableton Live, REAPER, Renoise oder auch Reason.

## Zusätzliche Hardware

Zahlreiche digitale Effektgeräte und Mischpulte sind heute ebenfalls über MIDI-Controller-Befehle steuerbar. Auf diese Weise können Aktivitäten am Pult mit einem Sequenzer oder einem PC aufgezeichnet werden, um z. B. einen Livemix nachzubearbeiten oder komplizierte Aktionen vorwegzunehmen. Auch sind Standardmischungen für bestimmte Aufnahmesituationen abspeicherbar. Größere Pulte der FOH sowie Monitormischpulte sind auf diese Weise durch Musiker fernsteuerbar.

# Protokoll

Das Protokoll wurde 1981 von Dave Smith für die Audio Engineering Society entwickelt und von der MIDI Manufacturers Association erstmals 1983 auf der NAMM-Show in Anaheim, USA, vorgestellt. Der MIDI-Standard wird von der MMA (MIDI Manufacturers Association) festgelegt.

Der folgende Abschnitt erfordert das Verständnis des Hexadezimalsystems. Ein Byte ist aus zwei Hexadezimalziffern zwischen 0 und 15 aufgebaut, wobei die Zahlen ab 10 mit A bis F notiert werden. Eine 2-stellige Hex-Zahl besitzt damit einen Zahlenbereich von 0 bis 255.

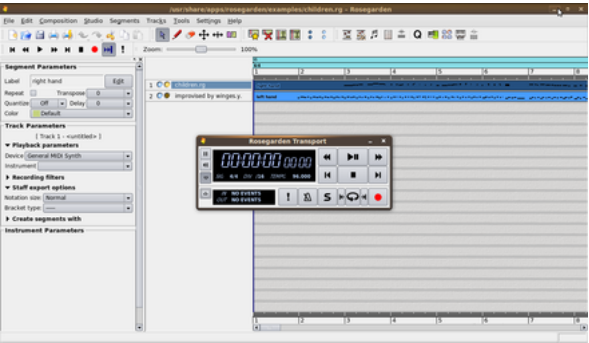
Die meisten MIDI-Befehle enthalten neben ihrer Befehlskennung und den Befehlsdaten auch eine Kanalnummer. Die Kanalnummer ist 4 Bit groß, so dass sich 2<sup>4</sup>, also 16 Kanäle ansteuern lassen. Jeder Kanal steuert ein spezielles Instrument, auch „Programm“ genannt.

## Nachrichtentypen

Das Statusbyte ist das erste Byte eines Befehls. Außerdem enthält das Statusbyte den betreffenden MIDI-Kanal *n*. Dieser reicht von 0 bis 15. In vielen Programmen wird bei der Darstellung der Kanalnummer die tatsächliche Kanalnummer um 1 erhöht dargestellt, also von 1 bis 16 statt von 0 bis 15. Die folgenden Bytes sind Datenbytes. Um einen unterbrochenen Datenstrom jederzeit korrekt wieder aufnehmen zu können beginnt ein Statusbyte stets mit

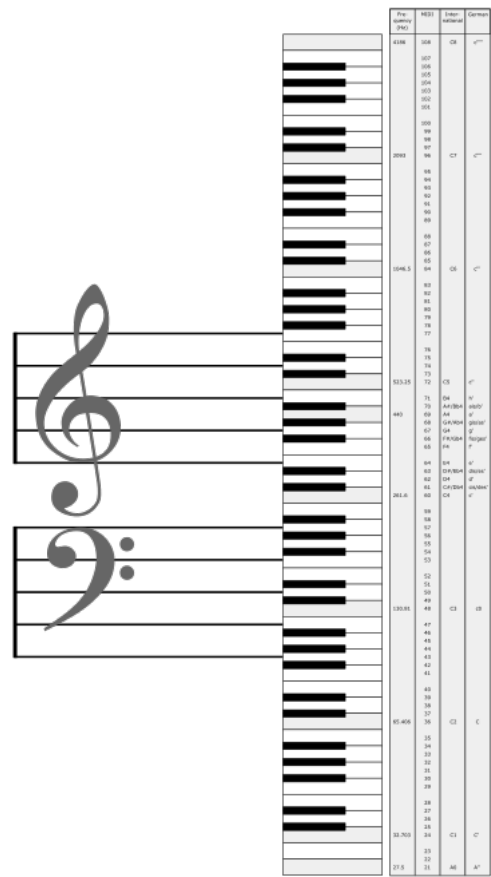


Hardwaresequenzer



Softwaresequenzer

einer 1 und ein Datenbyte mit einer 0. So liegen die Statusbytes im Bereich 80<sub>16</sub> bis FF<sub>16</sub> und die Datenbytes zwischen 00<sub>16</sub> und 7F<sub>16</sub>. Kommt anstelle eines erwarteten Statusbyte ein Datenbyte, dann gilt das letzte Statusbyte als wiederholt und das aktuelle Datenbyte zählt zu dessen Daten.



Notennamen und MIDI-  
Notennummern

Statusbyte und MIDI- Kanal Byte 0	Datenbyte Byte 1	Datenbyte Byte 2	Aktion	Erläuterung
<i>8n</i>	<i>kk</i>	<i>vv</i>	Note Off	Beendet das Spielen der Note <i>kk</i> . Hat also die Bedeutung des Loslassens einer Taste. Wurde die Note vorher gar nicht gespielt, wird dieses Signal einfach ignoriert. Zusätzlich wird eine „Release velocity“ <i>vv</i> gesendet. Im Dateiformat <b>.mid</b> kann dies, wenn <i>vv</i> weggelassen wird auch die Zeitspanne zwischen zwei Befehlen (bsp. Note an - Note aus) angeben, wobei hier ein Vierteltakt in 960 Einheiten aufgeteilt ist, die mit einem 14 Bit breiten Wert, dargestellt werden.
<i>9n</i>	<i>kk</i>	<i>vv</i>	Note On	Beginnt das Spielen einer Note <i>kk</i> . Zusätzlich wird die Anschlagsdynamik <i>vv</i> (engl. Velocity) angegeben, die der Druckstärke auf die Taste in 127 Schritten von 1 (sehr schwach) bis 127 (sehr stark) entspricht. Der Wert 0 ist als Note-Off-Befehl definiert.
<i>An</i>	<i>kk</i>	<i>vv</i>	Polyphonic Aftertouch	Beschreibt das Ändern des Tastendrucks <i>vv</i> während die Taste <i>kk</i> bereits gedrückt ist, für jede Taste einzeln. Diese Daten sind neutral, d. h., sie müssen anderen Daten zugeordnet werden. (z. B.: Zuordnung zu Controller 11: Expression – Ausdruckstärke, womit nach dem Anschlag der Taste über den „Bodendruck“ auf die Taste der Klang eines Saxophones verändert werden kann, während der Ton klingt.)
<i>Bn</i>	<i>cc</i>	<i>vv</i>	Control Change	Ändert den Zustand eines Controllers <i>cc</i> (siehe nächster Abschnitt) mit dem Wert <i>vv</i> .
<i>Cn</i>	<i>pp</i>		Program Change	Legt das für den angegebenen Kanal zu spielende Instrument <i>pp</i> fest.
<i>Dn</i>	<i>vv</i>		Monophonic bzw. Channel Aftertouch	Beschreibt das Ändern des Tastendrucks <i>vv</i> während die Tasten bereits gedrückt sind, für alle Tasten gemeinsam. Genau wie bei Polyphonic Aftertouch sind diese Daten neutral.
<i>En</i>	<i>vv</i>	<i>ww</i>	Pitch Bending	Einstellung des Pitchbend-Rades. Die zwei Datenbytes <i>vv</i> und <i>ww</i> ergeben zusammen den 14 Bit breiten Wert zwischen 0 und 16383. Zu beachten ist, dass sie vertauscht kombiniert werden, also (Byte 2 · 128) + Byte 1 = 14-Bit-Wert. Die Mittelstellung des Rades wird durch den Wert 8192 signalisiert.
<i>F0...7</i>	<i>xx</i>		System (exclusive) Message	Steuermeldungen, häufig gerätespezifisch, Länge ebenfalls gerätespezifisch ( <i>xx</i> = Datenbytes)

## Controller

Zweck eines MIDI-Controllers (Continuous Controller = CC, vordefinierter, festgelegter Controller) ist es, dem Anwender für sein Instrument die typischen Spielhilfen bereitstellen zu können, so z. B. dem Klavierspieler ein Haltepedal (CC064) oder dem Orgelspieler einen Lautstärkeregler (CC007 bzw. CC011) und einen Leslieschwindigkeitsumschalter (z. B. CC004, CC006).

Dem Synthesizerspieler beispielsweise stehen eine ganze Reihe weiterer Beeinflussungsregler für sein Spiel zur Verfügung, z. B. ein Modulationsrad (CC001), Pitch-Bending-Hebel, diverse Pedale und Fußschalter, weitere Schiebe- und Druckregler und -schalter, Blaswandler (CC002), Steuerungen per Licht für die Hände. Es gibt sogar – externe – Steuerungen für das Gehirn (Brain-to-Midi).

So kann eine gute Haptik und Kontrolle bei der Beeinflussung seiner Spielweise ein ausdrucksstarkes Spielen ermöglichen. Dazu haben die Spielhilfen meist mechanisch bewegliche Elemente, die der Benutzer bedienen kann. Diese Bewegung wird in Nachrichten (Controllerdaten) übersetzt und an die Geräte sowie Klangerzeuger weitergegeben. Sie arbeiten somit ähnlich wie Gamecontroller. Teilweise sind die Spielhilfen per MIDI-Protokoll an bestimmte Controllerbefehle gebunden, wobei meist sowohl die Spielhilfen beim Sender (z. B. Tastatur) wie auch beim Empfänger (Expandermodul, Plugin) frei auf andere Controller(befehle) umgeändert werden können (z. B. Modulationsrad als Lautstärkeregler).

Erst in neuerer Zeit nutzt man Controller verstärkt auch derart, dass die ursprüngliche Semantik ignoriert wird. Dem Befehl werden ganz andere Funktionen zugeordnet als im MIDI-Protokoll vorgesehen. Das ist häufig der Fall, wenn die Musikbearbeitung ausschließlich auf einem Computer durchgeführt wird. Insbesondere bei DJ-Programmen sind angepasste Controller sehr verbreitet. Hier wird zum Beispiel mit dem MIDI-Signal ein Dialog zur Songauswahl geöffnet, oder Funktionen wie Start, Stopp werden steuerbar. Auf der anderen Seite können Anzeigeelemente in Controllern bedient werden. Der semantisch mit der Tonhöhe belegte MIDI-Befehl kann so beispielsweise die Länge einer Wiederholungsschleife codieren. Gerade hier ist die Controllerhaptik wichtig, da sich mit der Maus am Computer die erforderliche Reaktionszeit und Feinfühligkeit zum Beispiel bei der Synchronisierung des Beats zweier Songs nicht erreichen lässt. Dem proprietären Charakter dieser Anwendungsart kann durch freie Belegbarkeit der MIDI-Befehle sowohl im Programm als auch im Controller begegnet werden, was allerdings noch lange nicht die Regel ist.

Klassische Anwendungen des MIDI-Controllers halten sich hingegen streng an die Semantik des Protokolls. Diese haben im Bereich der Musikerzeugung größere Bedeutung, da sich mit ihnen auf einfache Art und Weise gerätespezifische Klangparameter des aktuellen Instruments steuern lassen. Unterschiedliche Geräte sind dabei untereinander austauschbar.

Die Controller senden auf einem bestimmten Kanal mit einer bestimmten Controllernummer einen bestimmten Wert. Einfache Controller können Werte von 0 bis 127 annehmen, was jedoch bei Tonhöhenänderungen sehr schnell zu unschönen treppenhaft-sprunghaften Tonhöhenveränderungen führt. Daher lassen sich die Controller 0–31 mit einem sogenannten LSB-Controller 32–63 koppeln, um so eine wesentlich höhere Auflösung zu erhalten. In der Praxis wird diese Technik jedoch selten angewandt, da eine Auflösung der Lautstärke beispielsweise in 128 Schritten ausreichend ist.

Schalter wie beispielsweise das Haltepedal Nummer 64 können zwar theoretisch Werte zwischen 0 und 127 senden, da ein Schalter allerdings nur zwei Werte annehmen kann, werden üblicherweise Werte von 0 bis 63 als „aus“ und Werte von 64 bis 127 als „ein“ interpretiert.

Wird ein programmierbares Steuergerät verwendet, so sind Kenntnisse der Controllernummern und was diese üblicherweise steuern von großem Nutzen. Die wichtigsten Controller sind in der nachfolgenden Tabelle zusammengestellt. Das erste Byte eines Controllerbefehles lautet immer  $Bn_{16}$ , wobei  $n$  die Kanalnummer angibt und  $vv$  für den Wert, den der zu steuernde Klangparameter annehmen soll, steht.

## Herstellerspezifische Erweiterungen



Die allgemeinen (RPN - Registered Parameter Number) und die herstellerspezifischen (NRPN - Non Registered Parameter Number) Controller dienen dazu, Parameter zu steuern, die im normalen Controller-Bereich keinen Platz finden. Es können Parameter mit einem Index zwischen 0 und 16383 verändert werden. Zum Setzen der Indizes dienen die Controller

- RPN-MSB ( $65_{16}$ )
- RPN-LSB ( $64_{16}$ )
- NRPN-MSB ( $63_{16}$ )
- NRPN-LSB ( $62_{16}$ )

Danach wird entweder ein normaler MIDI-Controller ( $06_{16}$ ) mit einem Wert zwischen 0 und 127 gesendet oder ein Data Increment/Decrement ( $60_{16}$  oder  $61_{16}$ ), wobei hier das dritte Byte ignoriert wird. Wichtig ist, dass der MSB-Controller immer vor dem LSB-Controller versendet wird.<sup>[6]</sup> (siehe Tabelle).

## Systemexklusive Meldungen

*Systemexklusive Meldungen* (SysEx) sind Bestandteil des MIDI-Übertragungsprotokolles.

Während die übrigen MIDI-Befehle weitgehend als ein Standard festgelegt sind, wurde speziell durch die systemexklusiven Meldungen sichergestellt, dass die Hersteller von Hard- und Software auch Informationen übertragen können, die im MIDI-Protokoll nicht vorgesehen sind. So kann zum Beispiel der Speicherinhalt eines Gerätes zur Datensicherung an einen Rechner gesendet werden. Auch können Modellbezeichnungen zwischen den Geräten ausgetauscht werden. Das MIDI-Protokoll garantiert, dass systemexklusive Meldungen nicht durch andere Befehle unterbrochen werden. Daher werden systemexklusive Meldungen in Pausen übertragen, so dass es zu keinen unschönen Aussetzern oder Verzögerungen kommen kann.

Eine systemexklusive Meldung beginnt mit  $F0_{16}$  und endet mit  $F7_{16}$ . Das zweite Byte ist die Herstellerkennung, die von der MIDI Manufacturers Association vergeben wird und weltweit eindeutig ist.<sup>[7]</sup> Ist das zweite Byte gleich  $00_{16}$ , identifizieren die beiden folgenden Bytes den Gerätehersteller. Die Herstellerkennung ermöglicht den Geräten eine Filterung, was den entsprechenden Geräten die Interpretation der Meldungen ermöglicht. Zwischen der Herstellerkennung und dem Schlusszeichen folgen beliebig viele Datenbytes, die aber nur Werte zwischen  $00_{16}$  und  $7F_{16}$  enthalten können und so nur einen 7-Bit-Zeichensatz ermöglichen. In der Regel werden zu Beginn zuerst einige Bytes gesendet, die festlegen, für welches Gerät die Nachrichten bestimmt sind. Die Bedeutung legt der entsprechende Hersteller fest.

Beispiele:

- F0 41 ... F7
- F0 00 20 33 ... F7

## Synchronisation

Synchronisation ist immer dann nötig, wenn zwei oder mehrere MIDI- bzw. Audiogeräte zu einem System zusammengeschlossen werden müssen.

Dazu gibt es zwei MIDI-Protokolle:

Die ältere und tendenziell unpräzisere **MC (Midi-Clock)**, die auf das Grundtempo eines Songs referenziert und eher für den musikalischen Einsatz vorgesehen ist, und den neueren **MTC (Midi-Time-Code)**, der auf ein absolutes Zeitsystem referenziert und für Aufnahmesysteme relevant ist.

Jeder Systemverbund hat genau einen *Master*, alle anderen Geräte sind *Slaves*, die dem Master folgen müssen.

Synchronisation ist in vielen Bereichen notwendig. Dabei kann man zwei grundsätzlich verschiedene Einsatzzwecke feststellen, aus denen sich diverse Mischformen ergeben:

## 1. Synchronisation von Systemen

Hier werden verschiedene Aufnahme-, Wiedergabe- und Bearbeitungssysteme zu einer Systemeinheit verbunden. Sie sollen also zeitgleich und synchron starten, stoppen und mitlaufen und natürlich auch nicht während des Musikstücks auseinanderlaufen, was passieren würde, wenn man die einzelnen Systeme manuell gleichzeitig starten würde. Genau das übernimmt der MTC und verhindert derartige Probleme.

Einige Synchronisierungsbeispiele aus der Praxis (MTC):

- Zwei DAWs (Digital Audio Workstation) werden zu einem Verbund zusammengeschaltet.
- Eine DAW muss mit einem Hardware-Sequencer zusammengeschaltet werden.
- Eine DAW muss mit einer Drum-Machine zusammengeschaltet werden.
- Eine Bandmaschine muss mit einer DAW zusammengeschaltet werden.
- Eine Bandmaschine, ein Sequencer und ein Harddiskrecorder müssen zusammengeschaltet werden.
- Eine Bandmaschine oder eine DAW müssen für eine Mischung mit einem Mischpult mit eigenem Automationssystem zusammengeschaltet werden.

Bei (analogen) Bandmaschinen muss vor einer Aufnahme der Timecode auf das Band aufgespielt werden. Dabei wird ein SMPTE-Timecode auf die letzte Spur (8-Spur auf Spur 8, 16-Spur auf Spur 16 usw.) der Maschine aufgenommen, der von einem SMPTE-Generator erzeugt wird. Ein SMPTE-Code stört durch Übersprechen daneben liegende Kanäle, weshalb man auf die Außenspur ausweicht, um neben der SMPTE-Code-Spur nur noch eine weitere Spur für Aufnahmen zu verlieren.

Aus dem aufgenommenen SMPTE-Code „liest“ ein SMPTE-Reader das Signal aus und wandelt es in MTC, das von den daran angeschlossenen Geräten gelesen werden kann.

Da nur professionelle Maschinen aufwändige Synchronisatoren besitzen, sind meist die Bandmaschinen MTC-Master über den SMPTE-Code, denen alle anderen Systeme folgen müssen. Ein weiterer Grund ist, dass selbst professionelle Bandmaschinen immer eine Spulzeit haben, weshalb es selten einen Sinn ergibt, beispielsweise eine DAW als Master einzusetzen.

Je nach Art des SMPTE-Codes kann es durchaus einen Sinn ergeben, bei Überspielungen von Bandmaterial auch den alten Timecode vom Band mit als Audiosignal aufzunehmen, um Rechendifferenzen zu minimieren. Manchmal aber ist der Timecode bei alten Bändern so schlecht, dass eine Synchronisation nicht möglich ist, so dass alle Spuren mit einem Mal aufgenommen werden müssen. Den Timecode kann man nachträglich mit spezieller Software bzw. Geräten reparieren und wieder nutzbar machen.

## 2. Synchronisation von Synthesizerfunktionen und Effektgeräten

Hier werden Sequencer und Synthesizer bzw. Effektgeräte miteinander gleichgeschaltet, um musikalisch wichtige Ereignisse auszulösen.

Einige Synchronisierungs-Beispiele aus der Praxis (MC):

- Ein Delay soll genau auf den Rhythmus eines Musikstückes gesetzt werden, um ein musikalisches Echo für ein Soloinstrument bereitzustellen. Hierzu muss das Effektgerät natürlich MC verarbeiten können. Besonders wenn sich das Tempo des Stückes ändert, ist das nur auf diese Weise möglich, weil das manuell nicht umsetzbar wäre.

- Ein Flächensound eines Synthesizers soll rhythmisiert („zerhackt“) werden.
  - Entweder ein MIDI-Gate wird so angesteuert, dass es im Rhythmus des Musikstückes öffnet und schließt,
  - oder die MC wird zur Synchronisation der Lautstärkenmodulation im Synthesizer genutzt.
- Ein Sound eines Synthesizers soll im Rhythmus eines Musikstückes im Panorama wandern.
- Ein Sound eines Synthesizers soll im Rhythmus eines Musikstückes den Filter öffnen und schließen.

Die Möglichkeiten für den Einsatz von MC ist nur begrenzt durch die technischen Möglichkeiten der angeschlossenen Geräte.

## MIDI-Clock und Song Position Pointer

- MIDI-Clock (MC)

Die MIDI-Clock basiert auf Takt- und Notenebene. Die Einheit ist ein Tick, ein 96tel eines Schlages (bei vielen Liedern Viertelnote). Die Dichte, mit der die Clock-Informationen gesendet werden, ergibt sich daher aus der gewählten BPM-Einstellung des Liedes. Folgende Messages sind möglich:

1.  $F2_{16}$  Noten Position innerhalb eines Liedes 0–16383 (zwei Datenbytes)
2.  $F3_{16}$  Liedauswahl 0–127 (ein Datenbyte)
3.  $F8_{16}$  wird (während des Abspielens) 24 mal je viertel Note gesendet
4.  $FA_{16}$  Start
5.  $FB_{16}$  Von der aktuellen Position aus fortsetzen
6.  $FC_{16}$  Stopp

Durch diese Signale starten alle Sequenzer gleichzeitig, folgen dem gegebenen Tempo und stoppen auch wieder gleichzeitig.

- Song Position Pointer (SPP)

Während MC im Grunde lediglich Clock-Ticks sendet, ist SPP dafür zuständig, die Position im Song zu übermitteln. Er wird jede Sechzehntelnote übertragen und ist auf maximal 1024 Takte beschränkt.

Die relativ grobe Unterteilung von SPP-Befehlen dient zur groben Gleichschaltung von Geräten, während die exakte Gleichschaltung per MC erfolgt.

Angeschlossene Geräte können so auch im *Stopp-Modus* eines Sequenzers erkennen, an welcher Stelle in einem Musikstück sie sich befinden. Außerdem können sie bei bestimmten Songpositionen Funktionen ausführen, z. B. dass Drumcomputer vorprogrammierte Wiederholungen auslösen.

## MIDI-Time-Code

Der *MIDI timecode* (MTC) ist eine Umsetzung des SMPTE-Timecodes auf das MIDI-Format. Im Gegensatz zur MIDI-Clock ist der MTC eine reine Zeitinformation, eine Umrechnung auf Lied, Position innerhalb des Liedes und Abspieltempo muss durch die Software erfolgen.

Bei größeren Sprüngen der Position sendet der Master die absolute Position innerhalb einer SysEx-Message

F0
7F
7F
01
01
hh
mm
ss
ff
F7

1.  $F0_{16}$  SysEx Message Beginn
2.  $7F_{16}$  Dieser Herstellercode zeigt an, dass es eine universelle Realtime Message ist

3.  $7F_{16}$  Diese Kanalnummer zeigt eine Broadcast-Message an
4.  $01_{16}$  bezeichnet die Message als MIDI-Timecode
5.  $01_{16}$  zeigt eine volle, absolute timecode-message an
6. hh setzt sich zusammen aus SMPTE-Rate und Stunden in der Form  $0rrhhhh_2$ 
  1.  $hhhh_2$  Stunden von 0–23
  2.  $rr=00_2$  24 Bilder/s (Film)
  3.  $rr=01_2$  25 Bilder/s (PAL-Video)
  4.  $rr=10_2$  29,97 Bilder/s (drop frame timecode NTSC-Video)
  5.  $rr=11_2$  30 Bilder/s (non drop timecode NTSC-Video)
7. mm Minuten von 00–59
8. ss Sekunden 00–59
9. ff Einzelbild 00-nn, abhängig von der Frame-Rate
10.  $F7_{16}$  SysEx Message Ende

Während des Abspielens werden nur Short-Messages gesendet

```

F1  xx

```

Das Byte xx ist in der Bit-Darstellung am besten verständlich

1.  $0000\ 000y_2$  Frame Zähler low Nibble
2.  $0001\ yyyy_2$  Frame Zähler high Nibble
3.  $0010\ 00yy_2$  Sekunden Zähler low Nibble
4.  $0011\ yyyy_2$  Sekunden Zähler high Nibble
5.  $0100\ 00yy_2$  Minuten Zähler low Nibble
6.  $0101\ yyyy_2$  Minuten Zähler high Nibble
7.  $0110\ 0rry_2$  Stunden Zähler low Nibble + Framerate s. o.
8.  $0111\ yyyy_2$  Stunden Zähler high Nibble

Bei Rücklauf des Bandes kommen die Messages allerdings in umgekehrter Reihenfolge.

# Dateiformate

Für das Speichern von MIDI-Befehlen in *Standard-MIDI-Files* (kurz: SMF) gibt es drei Dateiformate:

- *SMF 0* – Beim Format 0 sind alle MIDI-Kanäle in einer Spur zusammengefasst. Dieses Format wird auch von Klingeltönen für Handys genutzt und kann mit gängigen Sequenzerprogrammen in das Format 1 konvertiert werden.
- *SMF 1* – Im Format 1 hat jeder Kanal seine eigene Spur und optional einen eigenen Namen. Verschiedene Stimmen und Instrumente können so besser identifiziert werden.
- *SMF 2* – Im Format 2 besteht jede Spur (Track) aus unabhängigen Einheiten. Im Gegensatz zu SMF 1 können also mehrere Spuren dieselbe MIDI-Kanal-Nummer haben.

Die standardisierte Dateiendung für MIDI-Dateien ist **.mid**. Daneben finden noch **.kar** Verwendung. Diese sogenannten Karaoke-Dateien enthalten den gesamten Liedtext. Das Dateiformat ist exakt dasselbe wie bei **.mid**. Viele Programme erkennen Dateien mit der Endung **.kar** jedoch nicht als MIDI-Dateien. (Die Datei kann aber in der Praxis einfach umbenannt werden.) Windows unterscheidet die Dateien, so dass erkennbar bleibt, dass es sich um eine Karaoke-Datei handelt. Die Datei kann trotzdem sowohl mit Karaoke-fähiger Software als auch mit normalen „Playern“ abgespielt werden. Microsoft benutzt die Endung **.rmi** für RIFF-RMID-Dateien. Bei diesen ist eine reguläre MIDI-Datei in einen RIFF-Container verpackt. RIFF-RMID ist kein offizieller MMA- oder AMEI-MIDI-Standard. Für Dateien, die MIDI-SysEx-Daten enthalten, wird **.syx** verwendet. Meist sind das Sound-Presets von Synthesizern.

# Alternative Übertragungswege

---

## USB und FireWire

Da MIDI im Wesentlichen ein Datenprotokoll zur Steuerung von elektronischen Musikinstrumenten darstellt, ist es prinzipiell unerheblich, über welche Hardware die Daten übertragen werden. Um eine kostengünstige, plattformübergreifende und vor allem schnelle Anbindung externer MIDI-Interfaces an einen Rechner zu erreichen, statten immer mehr Hersteller ihre Geräte neben den klassischen MIDI-Anschlüssen mit USB- oder FireWire-Anschluss (IEEE1394) aus. Dabei werden die MIDI-Befehle über USB bzw. FireWire getunnelt. Auf diese Art lassen sich mehrere virtuelle MIDI-Verbindungen realisieren, wodurch die Begrenzung auf 16 pro Verbindung praktisch keine Rolle mehr spielt. Diese Art von MIDI-Interfaces stellt die im Vergleich zum PC-Gameport deutlich zuverlässigere Variante zum Anschluss von MIDI-Geräten an den Rechner dar, da die verwendeten Treiber von den Herstellern dieser verhältnismäßig teuren Geräte zumeist auf Timinggenauigkeit hin optimiert werden. Für den professionellen Einsatz werden Interfaces mit vier bis acht einzeln adressierbaren Out-Ports verwendet, mit denen Timingprobleme deutlich vermindert werden können (vgl. auch folgender Absatz).

Wichtig ist zu wissen, dass sich das USB-MIDI-Protokoll vom herkömmlichen MIDI-Protokoll unterscheidet.<sup>[8]</sup> Die dort genannten „Jacks“, max. 16 pro USB-Endpoint, haben jeweils wieder 16 Kanäle.

## Wireless MIDI

Dem Trend hin zur drahtlosen Datenübertragung folgend werden auch Geräte angeboten, mit denen sich MIDI-Daten per Funk übertragen lassen. Die Geräte benutzen in der Regel Übertragungsfrequenzen im ISM-Band und senden bei Übertragungsfehlern ein „ALL NOTES OFF“, um hängende Töne zu vermeiden. Laut Herstellerangaben haben diese Geräte eine Reichweite von 10 bis 80 Metern.

## IP-basierende Netzwerke

Seit einiger Zeit gibt es eine Reihe von virtuellen Gerätetreibern, die es erlauben, MIDI-Daten über IP-basierende Netzwerke zu übermitteln. Während die meisten dieser Produkte auf proprietärer Basis die MIDI-Daten per TCP oder UDP über das Netzwerk übertragen, gibt es mittlerweile auch einen RFC für eine genormte Übertragung von MIDI-Daten über Netzwerke auf Basis des RTP-Protokolls: RFC 4695. Es gibt mehrere Open-Source-Implementierungen dieses Standards und auch das Apple-Netzwerk-MIDI von Mac OS X Tiger und iOS 4.2 basiert auf dieser Spezifikation. Es existiert auch ein Windows RTP-MIDI Treiber,<sup>[9]</sup> welcher auf Windows XP bis Windows 7 (32-Bit and 64-Bit) läuft und kompatibel zur Apple-Implementation ist.

## Andere

Bis vor Kurzem gab es Bemühungen, den von Yamaha entwickelten mLAN-Standard als Verknüpfung von MIDI- und Audiodaten auf der Basis von FireWire zu etablieren. Dies konnte sich jedoch nicht durchsetzen und wurde inzwischen eingestellt.

Eine weitere Option ist die Übertragung von MIDI-Daten über S/PDIF-Schnittstellen.

## Kritik

---



# Vor- und Nachteile gegenüber digitaler Audioaufzeichnung

Bevor MIDI entstand, war digitale Audioaufzeichnung noch extrem teuer und damit wenigen Produktionen vorbehalten. Somit eröffnete das technisch nicht sehr aufwendige MIDI mit seiner enormen Leistungsfähigkeit mittels Aufzeichnung reiner Steuersignale Anfang der 1980er Jahre einer breiten Masse von Musikschaaffenden plötzlich neue Horizonte. Dank MIDI können auch Amateurmusiker mit entsprechenden Kenntnissen komplexere musikalische Strukturen kreieren. Beispielsweise können Streicher und Bläser synthetisch imitiert werden, während man Schlagzeug, Gitarre und Gesang über Audiospuren einspielt.

MIDI-Signale enthalten lediglich Steuerdaten. Digitale „Audiosignale“ hingegen sind ein kontinuierlicher binärer Datenstrom, entstanden durch die sehr schnelle Abtastung (Digitalisierung) analoger Schwingungen einer Audioquelle (Mikrofon oder elektronisches Instrument). Sie haben eine konstant hohe Datenrate und können nach erfolgter Digital-Analog-Wandlung über ein Verstärker-Lautsprecher-System hörbar gemacht werden. MIDI-Daten fallen nur an, wenn Tasten auf einem Keyboard gedrückt oder losgelassen werden. So entstehen bei MIDI ungleich geringere Datenmengen. Aufgezeichnete MIDI-Signale können leicht im Nachhinein an einen anderen Klangerzeuger gesendet werden. Auch besteht die Möglichkeit, eingespielte MIDI-Daten im Nachhinein beliebig zu editieren, um etwa falsche Töne auf die richtige Tonhöhe oder Abspielposition zu bringen oder ihre Dynamik anzupassen. All diese Veränderungen der Originaleinspielung kosten im Vergleich zur Nachbearbeitung von digitalen Audioaufzeichnungen sehr wenig Rechenaufwand und sind mit allen heute verfügbaren Sequenzerprogrammen möglich.

## Latenz

Die Geschwindigkeit der MIDI-Schnittstelle ist vom technischen Standpunkt überholt. Da für das Auslösen eines Tons bis zu 3 Bytes übertragen werden müssen, was 0,960 ms in Anspruch nimmt, kommt es bei mehreren Events, die eigentlich gleichzeitig gesendet werden sollen, zu Verzögerungen, die bei Instrumenten mit deutlichem Anschlag auch hörbar werden können. Eine deutlich hörbare, durch MIDI verursachte Verzögerung tritt normalerweise auf, wenn zu viele MIDI-Daten auf einmal übertragen werden sollen, wie in den folgenden Beispielszenarien:

1. In Verbindung mit der Verwendung von System-Exclusive-Daten: Dieser Message-Typ belegt weit mehr als die typischen drei Bytes (1 Statusbyte und 2 Databytes).
2. Intensive Verwendung von Pitch-Bending oder ähnlichen Controllern.
3. Bei der Verkettung mehrerer Hardware-Klangerzeuger über MIDI-Out oder software-gepufferte MIDI-Thru-Verbindungen: Hier entsteht jeweils eine weitere Verzögerung von mindestens einem Byte.

Bei Studioarbeit werden dagegen mehrere Maßnahmen ergriffen:

1. Vermeidung der genannten Belastungen
2. Verteilung der MIDI-Daten auf mehrere parallele Ausgänge eines oder mehrerer MIDI-Interfaces.
3. Besonders zeitkritische Synchronisierungsdaten wie MTC oder MIDI-Clock werden auf eigenen MIDI-Verbindungen übertragen.
4. „Vorziehen“ des MIDI-Signalfusses: Man spielt Teile der MIDI-Daten etwas vorgezogen ab.
5. Nutzung von Funktionen im Sequenzer, um die Daten auszudünnen. Auf diese Weise werden redundante MIDI-Daten ausgefiltert und gelangen nicht in den MIDI-Datenfluss.

Bei Live-Anwendungen sind diese Maßnahmen allerdings oft nicht möglich. Ein Masterkeyboard hat üblicherweise nur ein MIDI-Out, das heißt, alles, was dort an Noten und Controllern ausgelöst wird, muss über diese eine Schnittstelle.

Trotz dieser Einschränkungen und des hohen Alters erfreut sich MIDI nach wie vor großer Beliebtheit, da es weit verbreitet, gut standardisiert und sehr zuverlässig ist. Eine Fortentwicklung wäre aus technischer Sicht vielleicht wünschenswert, hätte aber für den normalen Anwender mehr Komplikationen als Vorteile. Eine neue MIDI-

Spezifikation würde möglicherweise alle vorher produzierten Instrumente inkompatibel machen. Weiterhin ist zu bedenken, dass heute vornehmlich mit Plugins in der Musikproduktion gearbeitet wird, wo das MIDI-Timing kein Problem darstellt.

Die Präzision einer MIDI-Sequenz ist auf 1/96 einer Viertelnote definiert (siehe MIDI Clock), in einigen Fällen können Sequenzerprogramme eine noch höhere Genauigkeit erreichen (bei der Verwendung von Plugins). Das bedeutet, dass die kleinste reproduzierbare Timingschwankung vom Songtempo abhängig ist. In einem Lied mit 120 bpm beträgt dieser Wert 5,21 Millisekunden.

Von Yamaha gibt es mit XG-MIDI eine Erweiterung, die genau wie Rolands GS-MIDI Verbesserungen in der Kompatibilität von Standard-MIDI-Files bringt, allerdings nicht über ein proprietäres System hinaus gedieh. Als Quasi-Standard durchgesetzt hat sich lediglich GM (General MIDI). Beide Standards nutzen das normale MIDI-System ohne Änderungen an der MIDI-Hardware oder dem Protokoll.

## Stimmungen und Skalen

Schwerer wiegen die Einschränkungen musikalischer Art: MIDI wurde für die Steuerung von Synthesizern konzipiert, die überwiegend in derselben temperierten Stimmung betrieben werden. Für nicht-temperierte Stimmungen oder andere Skalen müssen die Notenwerte entweder per Controller im Sequenzer künstlich modifiziert oder in den Endgeräten uminterpretiert werden. Viele hochwertige Keyboards verfügen inzwischen über eine Stimmungseinstellung, die das bewerkstelligt. Für komplexere Stimmungprobleme und alle anderen Geräte existiert spezielle Software (sog. Mikrotuner). Diese setzt wiederum eine MIDI-Schnittstelle (z. B. USB-MIDI-Adapter) am Rechner voraus.

## Auflösung

Alle analogen und kontinuierlichen Aspekte wie die Anschlagdynamik, das Verhalten des Klangs während des Drückens (after touch) und nach dem Loslassen der Taste sowie die Echtzeitmodifikationen der Amplitude, der Frequenz etc. sind in den 128 Stufen des MIDI-Formats aufgelöst. Das schafft unweigerlich eine verfälschende Vergröberung des manuellen Spiels schon bei der Aufnahme und verhindert eine stufenlose Änderung, was vor allen bei der Lautstärke und dem Spieltempo problematisch ist. Zwar werden in vielen modernen Keyboards, Klangerzeugern und Synthesizern die ankommenden Controllerwerte geglättet und sanft auf neue Werte übergegangen, dennoch hat damit der Musiker nur eine eingeschränkte Kontrolle über den Klang.

Ein reales Instrument verfügt über keinerlei Abstufung in Tonhöhe oder Lautstärke. Selbst bei der Steuerung von künstlichen Klangparametern bei Synthesizern, wie z. B. der Cutoff-Frequenz eines Filters, macht sich diese „Stufigkeit“ teilweise hörbar bemerkbar.

Auch Instrumente mit scheinbar festen Tonhöhen wie Flöten oder Klaviere besitzen eine gewisse Varianz in der Tonhöhe während des Spiels, die bei Blasinstrumenten vom Luftstrom und bei Saiteninstrumenten vom Anschlag und der Momentanlautstärke sowie allerlei Resonanzeffekten abhängt. Die während eines komplexen Tonverlaufes existierenden Schwankungen der Tonparameter Amplitude und Phase sind daher nur sehr grob von MIDI-Hard- und Software nachzubilden bzw. zu erzeugen. Bei anschlagsdynamischen Instrumenten macht sich dieses besonders bemerkbar (Bösendorfer 290SE). Auch Vibrato und Blue-Note-Technik kann kaum korrekt abgebildet werden.

Dieses Problem kann nur durch die Nutzung zweier aufeinander folgender MIDI-Controller umgangen werden, was den zeitlichen Rahmen weiter einschränkt. Ein anderes Beispiel für die Benutzung zweier verknüpfter Controller ist der Bank Select-Befehl (MIDI CC 0 und 32 des General-MIDI-Standards), der es erlaubt, bis zu 128 Bänken zu

adressieren. Jede Bank kann wiederum Subbänke (max. 128) enthalten. Auf diese Weise können theoretisch 16384 Soundbänke, jede einzelne mit 128 Klängen, realisiert werden. Spezielle Klangbibliotheken bieten deshalb verschiedene Artikulationen und Spielweisen akustischer Instrumente an, um diesen Nachteil zu umgehen.

## Literatur

---

- Christian Braut: *Das MIDI-Buch*. Sybex, Düsseldorf u. a. 1993, [ISBN 3-8155-7023-9](#).
- Bernd Enders, Wolfgang Klemme: *Das MIDI- und SOUND-Buch zum ATARI ST. MIDI-Grundlagen, MIDI-Befehle, Programmierung des Soundchips, Steuerung von MIDI-Instrumenten. Alles über professionelle Musiksoftware, MIDI- Programmtools, Umrechnungstabellen*. Markt & Technik Verlag AG, Haar bei München 1988, [ISBN 3-89090-528-5](#).
- Erol Ergün: *Cubase SX/SL in der Praxis. Die neue Generation der MIDI/Audio-Musikproduktion. Einstieg, Profi-Tipps und Strategien für die Musikproduktion mit Steinberg Cubase SX/SL*. 3., aktualisierte und erweiterte Auflage. PPV, Presse-Project-Verlags-GmbH, Bergkirchen 2005, [ISBN 3-937841-22-9](#).
- Dieter Stotz: *Computergestützte Audio- und Videotechnik. Multimediatechnik in der Anwendung*. 2. Auflage. Springer, Berlin u. a. 2011, [ISBN 978-3-642-23252-7](#).

## Weblinks

---

 **Commons: MIDI (<https://commons.wikimedia.org/wiki/Category:MIDI?uselang=de>)** – Sammlung von Bildern, Videos und Audiodateien

- MIDI Manufacturers Association (<http://www.midi.org/>)
- Die komplette MIDI-1.0-Detail-Spezifikation von 1996 und Erweiterungen ([https://midi.org/images/downloads/complete\\_midi\\_96-1-3.pdf](https://midi.org/images/downloads/complete_midi_96-1-3.pdf))
- Der Midi-Guide (<http://unseretollepage.de/hosted/midiguide/index.html>)
- Deutschsprachige Seite zur MIDI-Spezifikation (<http://home.snafu.de/sicpaul/midi/midi0a.htm>)
- USB-MIDI-Spezifikation für Entwickler ([http://www.usb.org/developers/docs/devclass\\_docs/midi10.pdf](http://www.usb.org/developers/docs/devclass_docs/midi10.pdf)) (englisch; PDF; 180 kB)
- RTP-MIDI-Spezifikation – RFC 4695 (<http://tools.ietf.org/html/rfc4695>)
- MIDI-Klavatur, Frequenzen, Notennamen und Notenummern (<http://www.sengpielaudio.com/Rechner-notennamen.htm>) (private Seite)
- MIDI-Basics (<http://www.bonedo.de/artikel/einzelansicht/midi-basics.html>) auf bonedo.de
- Special: 30 Jahre MIDI – Galaktisch gut ([http://www.tastenwelt.de/30\\_jahre\\_midi.0.html](http://www.tastenwelt.de/30_jahre_midi.0.html)), tastenwelt.de (3/2013)
- Ratgeber zu Midi Keyboards (<http://www.midi-keyboard.net>) Bauweisen, Einsatzbereiche, Software und Verbindungsmöglichkeiten

## Quellen

---

1. Seite des MMA (Midi Manufacturers Association) (<http://www.midi.org/techspecs/index.php>)
  2. [http://educyclopedia.karadimov.info/library/The\\_MIDI\\_Specification.pdf](http://educyclopedia.karadimov.info/library/The_MIDI_Specification.pdf)
  3. [http://www.doc.ic.ac.uk/~nd/surprise\\_97/journal/vol1/aps2/](http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol1/aps2/)
  4. <http://www.bonedo.de/artikel/einzelansicht/podcast-der-erfinder-von-midi-dave-smith.html>
  5. *MIDI-1.0-Electrical-Specification-ca33*. (<https://mitxela.com/other/ca33.pdf>) AMEI (Association of Musical Electronics Industry), 16. September 2015, abgerufen am 19. August 2017.
  6. <http://musik-steiermark.at/musiktechnik/midi/nachrichten.htm>
  7. <http://www.midi.org/techspecs/manid.php>
  8. Universal Serial Bus Device Class Definition for MIDI Devices ([http://www.usb.org/developers/docs/devclass\\_docs/midi10.pdf](http://www.usb.org/developers/docs/devclass_docs/midi10.pdf)) (PDF; 180 kB) (englisch), Kapitel 4: *USB-MIDI Event Packets*
  9. Windows RTP-MIDI Treiber (<http://www.tobias-erichsen.de/rtpMIDI.html>)
- MIDI Messages (<http://www.midi.org/techspecs/midimessages.php>)

**Diese Seite wurde zuletzt am 9. Oktober 2017 um 13:22 Uhr bearbeitet.**

Der Text ist unter der Lizenz „[Creative Commons Attribution/Share Alike](#)“ verfügbar; Informationen zu den Urhebern und zum Lizenzstatus eingebundener Mediendateien (etwa Bilder oder Videos) können im Regelfall durch Anklicken dieser abgerufen werden. Möglicherweise unterliegen die Inhalte jeweils zusätzlichen Bedingungen. Durch die Nutzung dieser Website erklären Sie sich mit den [Nutzungsbedingungen](#) und der [Datenschutzrichtlinie](#) einverstanden. Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.