

MIDI Datei Format

MIDI steht für *Musical Instrument Digital Interface* und ist ein weit verbreiteter Standard zum Übertragen von Musiksteuerzeichen. Das Datenübertragungsprotokoll kann dabei in Form einer Datei auch gespeichert, editiert und abgespielt werden. Gegenüber dem MP3 oder WAV Format bietet es den Vorteil, Musikstücke in geringer Datengröße zu speichern ohne dabei Verluste aufgrund von Kompressionsalgorithmen in Kauf nehmen zu müssen.

Neben Angaben über Musiknoten können auch Metadaten wie bspw. Interpret, Copyright usw. oder auch Lyrics für Karaoke gespeichert werden.

Das MIDI Datei Format ist ein sehr anspruchsvolles Format. Informationen in deutsch sind rar. Deswegen entstand dieses Dokument. Es kann aber auch nicht auf alle Raffinessen eingehen.

Aufbau

Eine MIDI Datei besteht aus 2 Arten von Chunks: Einem Header-Chunk *MThd* und einem oder mehreren Track-Chunks *MTrk*. Die Daten werden im BigEndian Format gespeichert. Erschwerend kommt hinzu dass oft mit 7 Bit Daten gearbeitet wird und einige Längenangaben variabler Länge sind.

Ich werde folgende Notation für die Datentypen verwenden

Datentyp	Größe in Bytes	Wertebereich
Byte	1	0 .. $2^8 - 1$
Short	2	0 .. $2^{16} - 1$
Int	4	0 .. $2^{32} - 1$

Header-Chunk

Der Header-Chunk hält grundlegende Informationen über das Musikstück bereit. Er gibt an, wie viel Tracks sich in der Datei befinden, wie diese abgespielt werden und wie Zeitwerte zu interpretieren sind.

Name	Datentyp	Beschreibung
Id	Int	„MThd“ bzw. 0x4D546864
Length	Int	6 bzw. 0x00000006
Format	Short	0: Single Track 1: Multiple Tracks synchronous 2: Multiple Tracks asynchrhonous
NumTracks	Short	Anzahl der Tracks
TimeDevision	Short	Siehe Text

Anhand des `Id`-Feldes kann man eine MIDI Datei identifizieren. Wohlgermerkt sei aber hier, dass MIDI Dateien häufig auch im RIFF-Container Format untergebracht werden.

Die Länge des Header-Chunks beträgt ausschließlich 6 Byte. Ist eine Länge > 6 Byte angegeben, so kann man die übrigen Bytes überspringen oder eine Fehlermeldung ausgeben(ich bevorzuge letzteres).

Das `Format`-Feld gibt an, wie die einzelnen Tracks abgespielt werden. Wurde `Single Track` gewählt, sollte die Datei nur einen einzigen Track enthalten(`NumTracks = 1`).

Beim `Multiple Tracks synchronous` Format werden alle vorhandenen Tracks gleichzeitig abgespielt. In der Regel spricht jeder Track nur einen Kanal an(von übrigens 16 möglichen).

Wurde `Multiple Tracks asynchronous` gewählt, so werden die Tracks abhängig von ihrer Sequenznummer abgespielt

Wie viel Tracks auf den Header-Chunk folgen bestimmt das Feld `NumTracks`. Die Anzahl sollte in Abhängigkeit des Formats sein. Also bei `Single Track = 1`, bei `Multiple Tracks > 1`.

Das Feld `TimeDivision` sorgt dafür, dass Zeitangaben korrekt interpretiert werden können. Es gibt zwei Formate für dieses Feld. Unterscheiden kann man diese anhand des ersten Bits. Ist dieses nicht gesetzt, wird im PPQN- andernfalls im SMPTE Format gearbeitet.

Name	Aufbau	Beschreibung
PPQN	0XXXXXXXX XXXXXXXX	Ticks / Beat oder Pulses / Quarter Note Wie lang eine ¼ Note ist.
SMPTE	1XXXXXXXX YYYYYYYY	Ticks / Frames / Second XXXXXXXX Frames / Second 24, 25, 29 oder 30 YYYYYYYY Ticks / Frame

Zugegeben ist das recht kompliziert gemacht. Wir müssen wissen, dass im Track-Chunk ausschließlich MIDI Events zu finden sind. Dieses Events haben einen zeitlichen Abstand voneinander. Dieser Abstand wird im `DeltaTime` Feld angegeben(später dazu mehr).

Ein einfaches Beispiel:

NoteOn

```
- DeltaTime: 0
- Channel: 1
- NoteNumber: 24
- Velocity: 100
```

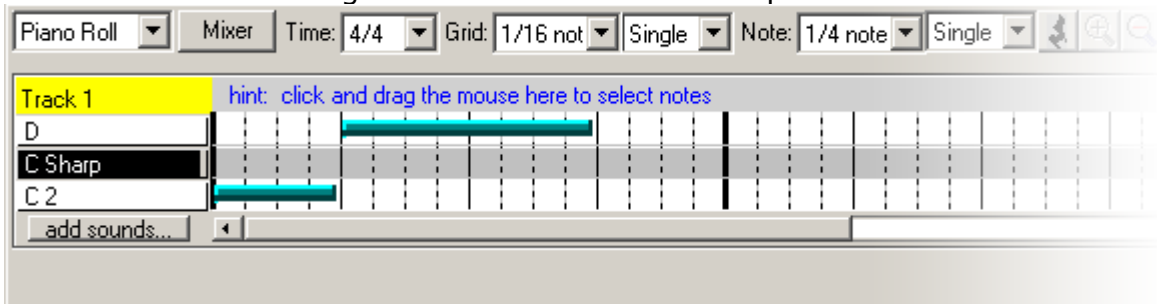
NoteOff

```
- DeltaTime: 480
- Channel: 1
- NoteNumber: 24
- Velocity: 0
```

```
NoteOn
- DeltaTime: 0
- Channel: 1
- NoteNumber: 26
- Velocity: 100
```

```
NoteOn
- DeltaTime: 960
- Channel: 1
- NoteNumber: 26
- Velocity: 0
```

In unserem Beispiel soll `TimeDevision` im PPQN Format vorliegen und eine $\frac{1}{4}$ Note soll 480 MIDI Ticks lang sein. Resultat unseres Beispiels:



Anvil Studio – Piano Roll Editor; `NoteNumber` 24 entspricht C in der zweiten Oktave(C2) und `NoteNumber` 26 D in der zweiten Oktave(D2).

Mit dem ersten `NoteOn` Event drücken wir die C Taste auf dem Klavier. Nach 480 MIDI Ticks folgt dann ein `NoteOff` Event – die Taste wird wieder losgelassen. Gleich im Anschluss – ohne Zeitverzögerung – folgt ein weiteres `NoteOn` Event mit der Taste D. Nach 960 MIDI Ticks wird auch diese wieder losgelassen.

Im Header haben wir `TimeDevision` mit 480 PPQN angeben. Also wird beim ersten mal eine $\frac{1}{4}$ Note gespielt, beim zweiten mal eine $\frac{1}{2}$ Note gespielt. Würden bis zum `NoteOff` Event nur 240 Ticks vergehen, würde eine $\frac{1}{8}$ Note gespielt.

Und tatsächlich verrät uns der Blick in den Composer, dass diese beiden Noten gespielt werden:

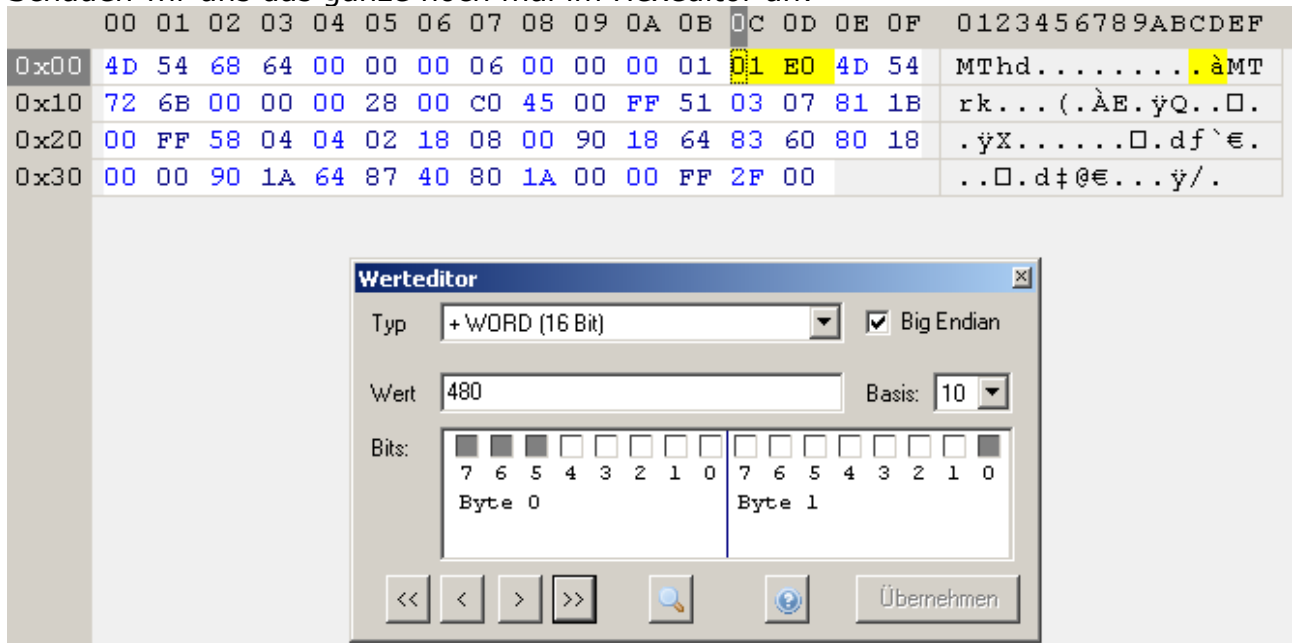


Dies besagt jedoch noch nicht, wie lange eine $\frac{1}{4}$ Note gespielt wird. Dazu dient bspw. das `SetTempo` Event – es gibt an, wie viel Millionstelsekunden eine $\frac{1}{4}$ Note einnimmt. Doch später dazu mehr.

Unser Beispiel Header sieht folgendermaßen aus:

Id: MThd
Length: 6
Format: Single Track
NumTracks: 1
TimeDivision: 0x01E0

Schauen wir uns das ganze noch mal im Hexeditor an:



Musiktheorie

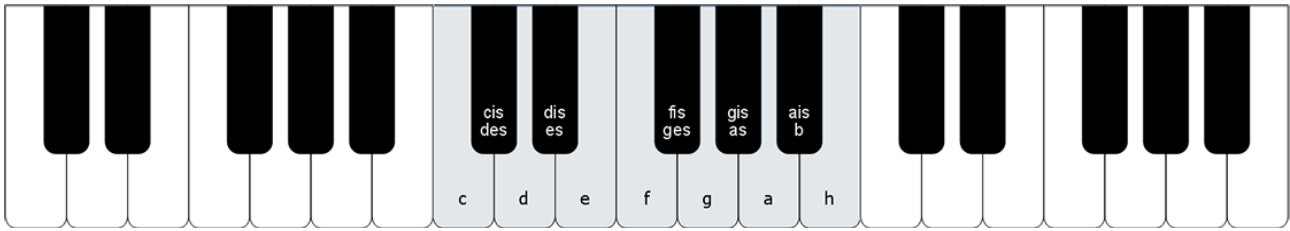
Wir haben es hier mit Instrumenten zu tun also müssen wir auch Hintergrundwissen dazu haben. Es folgt eine kleine Einführung in die Theorie für die Nicht-Tontechniker und Musiker unter uns (mich mit eingeschlossen)

Die Definition des Tons kann variieren. Für die meisten ist ein Ton eine sinusförmige Schalldruckwelle - Sinuston. Unterscheiden lassen sich Töne in ihrer Tonhöhe. Die Amplitude der Sinuskurve gibt die Lautstärke, die Periodendauer die Tonhöhe/Tonfrequenz in Hz an.

Mehrere Töne übereinander gelagert ergibt ein Geräusch. Sind diese Töne harmonisch zueinander ist das Geräusch ein Klang. Bspw. erzeugt ein Schlagzeug ein Geräusch hingegen ein Klavier einen Klang.

Klänge haben keinen oder mehrere Obertöne, einen Grundton und keinen oder mehrere Untertöne. Dabei nehmen Obertöne ein Vielfaches und Untertöne ein Bruchteil der Frequenz des Grundtons ein.

Eine Oktave besteht aus 7 Stammtönen (C, D, E, F, G, A, H) und 5 Halbtönen (Cis, Dis, Fis, Gis, Ais).



Das MIDI Protokoll umfasst 128 Noten:

Nummer	Name	Nummer	Name	Nummer	Name	Nummer	Name
0	C-2	32	G#0	64	E3	96	C6
1	C#-2	33	A0	65	F3	97	C#6
2	D-2	34	A#0	66	F#3	98	D6
3	D#-2	35	B0	67	G3	99	D#6
4	E-2	36	C1	68	G#3	100	E6
5	F-2	37	C#1	69	A3	101	F6
6	F#-2	38	D1	70	A#3	102	F#6
7	G-2	39	D#1	71	B3	103	G6
8	G#-2	40	E1	72	C4	104	G#6
9	A-2	41	F1	73	C#4	105	A6
10	A#-2	42	F#1	74	D4	106	A#6
11	B-2	43	G1	75	D#4	107	B6
12	C-1	44	G#1	76	E4	108	C7
13	C#-1	45	A1	77	F4	109	C#7
14	D-1	46	A#1	78	F#4	110	D7
15	D#-1	47	B1	79	G4	111	D#7
16	E-1	48	C1	80	G#4	112	E7
17	F-1	49	C#1	81	A4	113	F7
18	F#-1	50	D1	82	A#4	114	F#7
19	G-1	51	D#1	83	B4	115	G7
20	G#-1	52	E2	84	C5	116	G#7
21	A-1	53	F2	85	C#5	117	A7
22	A#-1	54	F#2	86	D5	118	A#7
23	B-1	55	G2	87	D#5	119	B7
24	C0	56	G#2	88	E5	120	C8
25	C#0	57	A2	89	F5	121	C#8
26	D0	58	A#2	90	F#5	122	D8
27	D#0	59	B2	91	G5	123	D#8
28	E0	60	C3	92	G#5	124	E8
29	F0	61	C#3	93	A5	125	F8
30	F#0	62	D3	94	A#5	126	F#8
31	G0	63	D#3	95	B5	127	G8

Wenn man ein Musikinstrument stimmt, benutzt man dazu eine Stimmgabel die einen Sinuston mit 440 Hz erzeugt. Das ist der Kammerton und ist mit der Notenummer 69 als A3 verzeichnet.

Ausgehend davon lassen sich die Frequenzen für alle anderen Noten folgendermaßen berechnen:

Die Note C-2 mit der Nummer 0 ist definiert als:

$$c * 2^{\frac{9}{12}} \text{ Hz} = \frac{440}{32} \text{ Hz}; \quad c * 2^{\frac{3}{4}} \text{ Hz} = \frac{55}{4} \text{ Hz}$$

$$c = \frac{\frac{55}{4} \text{ Hz}}{\frac{2^{\frac{3}{4}}}{1} \text{ Hz}}; \quad c = \frac{55}{4 * 2^{\frac{3}{4}}}; \quad c \approx 8,173$$

Alle anderen Tonhöhen sind durch

$$f = c * 2^{\frac{n}{12}} \text{ Hz}$$

definiert. Dabei ist c die oben angegebene Konstante für C-2 mit der Notennummer 0 und n die Notennummer von der man die Frequenz wissen möchte. Die Note G#0 mit der Notennummer 64 hat bspw. die Frequenz

$$f = c * 2^{\frac{64}{12}} \text{ Hz}; \quad f \approx 8,173 * 2^{\frac{16}{3}} \text{ Hz}; \quad f \approx 329,628 \text{ Hz}$$

. Und auch für den Kammerton A mit der Nummer 69 funktioniert diese Formel

$$f = c * 2^{\frac{69}{12}} \text{ Hz}; \quad f \approx 8,173 * 2^{\frac{69}{12}} \text{ Hz}; \quad f \approx 440,000 \text{ Hz}$$

. Die Formel besagt auch, dass sich die Frequenz eines Tons in der nächst höheren Oktave verdoppelt und in der nächst kleineren Oktave halbiert. So hat A3 in der 4ten Oktave 880 Hz und in der 2ten Oktave nur 220 Hz.

Notenwerte geben die relative Tondauer an. Wie lange sie tatsächlich gespielt wird, wird vom Dirigenten bzw. in unserem Fall vom `SetTempo` Event bestimmt.



Von links nach rechts: Ganze Note, 1/2 Note, 1/4 Note, 1/8 Note, 1/16 Note, 1/32 Note, 1/64 Note, 1/128 Note, 1/256 Note, 1/512 Note

Darunter die Pausen mit gleicher Länge.

Bei punktierten Noten erhöht sich der Notenwert um die Hälfte. Bei einer punktierten 1/8 Note bspw. erhöht sich der Notenwert um 1/16 auf 3/16.

Bei 480 PPQN ergibt sich folgende Tabelle:

Notenwert	Ticks	Notenwert	Ticks
1	1920	3/2	2880
1/2	960	3/4	1440
1/4	480	3/8	720
1/8	240	3/16	360
1/16	120	3/32	180
1/32	60	3/64	90
1/64	30	3/128	45
1/128	15	3/256	--
1/256	--	3/512	--
1/512	--	3/1024	--

Zusammenfassend sollte man über Notennamen und Notenwerte bescheidwissen. Ein Musiker wird sich wohl weniger mit Frequenzen und Dezibel Verhältnissen auseinander Setzen - wer aber vor hat, MIDI Abspielprogramme zu entwickeln, muss sich auf diesem Gebiet nochmal schlau machen.

Track-Chunk

Der Track-Chunk beinhaltet Informationen wie etwa den Tracknamen, Tempoangaben, Lyrics uvm. Diese Angaben sind in einer Abfolge von Events gespeichert. Unterteilt werden diese Events in Channel-, Meta- und SysEx Events.

Name	Datentyp	Beschreibung
Id	Int	„MTrk“ bzw. 0x4D54726B
Length	Int	Länge der Daten. Die Felder Id und Length werden nicht mit einbezogen!
Events	-	Abfolge von Events

MIDI-Events

Die MIDI Events sind Steueranweisungen. Dabei ist ihre Reihenfolge zeitlich geregelt. Es gibt 3 Arten von MIDI Events:

Name	Beschreibung
Channel Events	NoteOff, NoteOn, PitchBend etc.
Meta Events	InstrumentName, Lyrics, SetTempo etc.
SysEx	System Exclusive - Hersteller abhängig

Unabhängig vom Typ besitzt jedes Event das Feld `DeltaTime`.

Name	Beschreibung
DeltaTime	Die Zeitdifferenz zum vorhergehenden Event. Es ist in variabler Länge encodiert.
EventData	Hier stehen die Event spezifischen Daten. Unterscheiden lassen sich alle Events vor allem durch das erste Byte.

Eine Deltzeit von 0 gibt an, dass dieses Event simultan(zeitgleich) zum vorhergehenden Event gespielt werden sollte.