



MIDI

Warum sich Computer und elektronische Musikinstrumente gut verstehen

Eine komplizierte Beziehung mit Happy-End

Martin Roppert, 18. November 2021



MIDI – Computer und Musik

- MIDI-Standard, Begriffsklärung und Geschichte
- Wie es funktioniert
- Regeln und Beispiele

MIDI-Standard, Begriffsklärung und Geschichte

- MIDI steht für „Musical Instrument Digital Interface“
 - Elektronische Instrumente kommunizieren miteinander
 - Austausch von „Spielinformationen“
 - Festgelegt in „MIDI 1.0 Detailed Specification“ (www.midi.org)
- Mittlerweile auch
 - Audio Mixing, Editieren, Produktion
 - Spiele, Robotik, (Bühnen-)Beleuchtung, Klingeltöne
- MIDI überträgt Informationen, Beschreibungen, Kommandos
- MIDI überträgt NICHT Musik

MIDI-Standard, Begriffsklärung und Geschichte

- 1940er – erste elektronische Musikinstrumente
- 1960er – erste Synthesizer
 - Roland, Oberheim, Sequential Circuits und Fender Rhodes haben proprietäre Standards für die Steuerung ihrer Instrumente entwickelt
- 1980er – erste Standards
 - Sequential Circuits: Technical Paper (USI: Universal Synthesizer Interface) im Oktober 1981 der Audio Engineering Society (AES) vorgelegt
 - Ein Jahr lang Abstimmung zwischen verschiedenen Herstellern
 - Erster MIDI-Synthesizer “Prophet 600” von Sequential im Dezember 1982
 - Einen Monat später Verbindung “Prophet 600” mit “Roland Jupiter 6”
 - Offizielle MIDI-Spezifikation 1.0 im August 1983



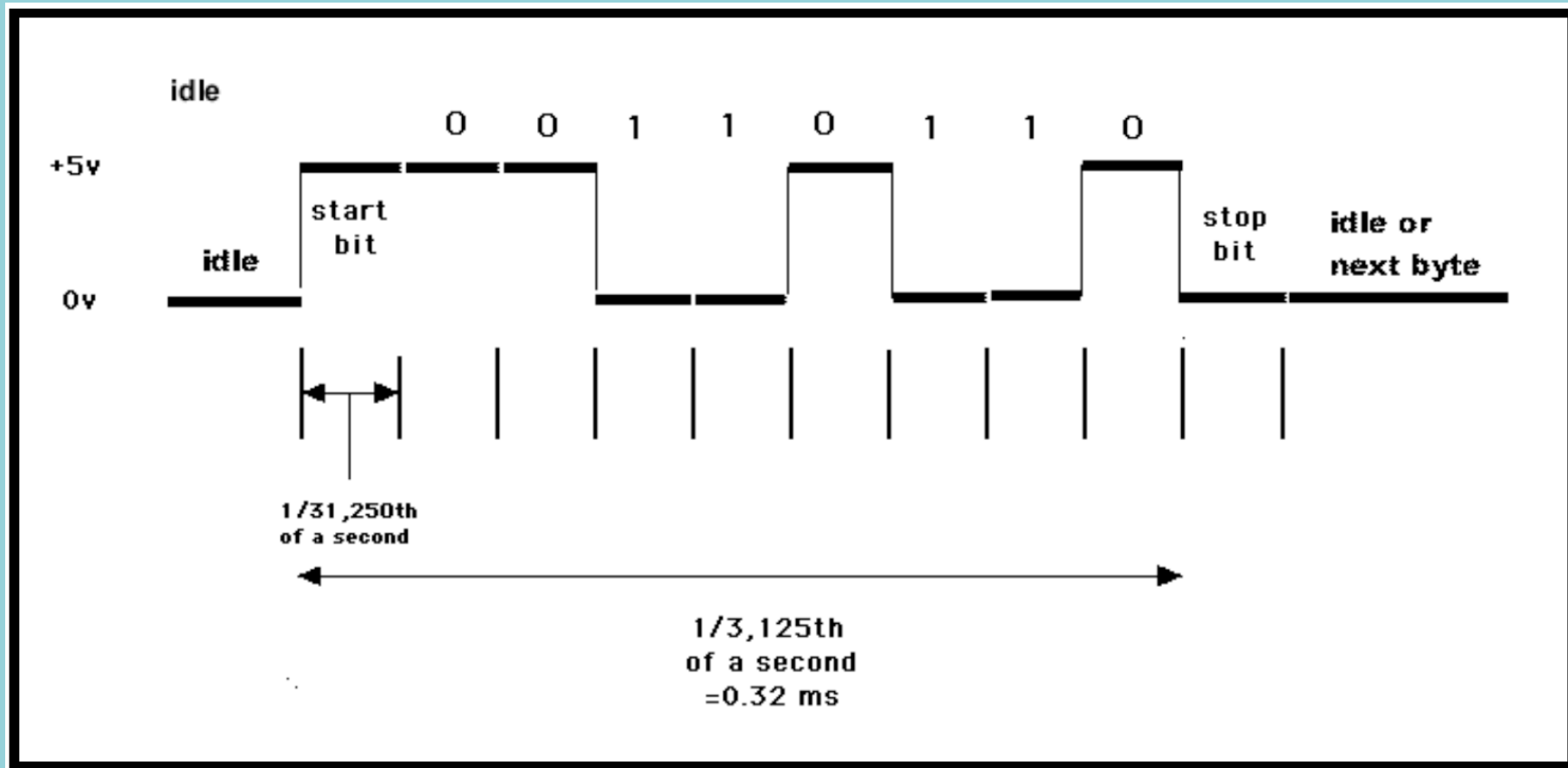
The Prophet 600 and the Jupiter 6 at the 1983 Winter NAMM show

Quelle: <https://www.midi.org/articles/a-brief-history-of-midi>

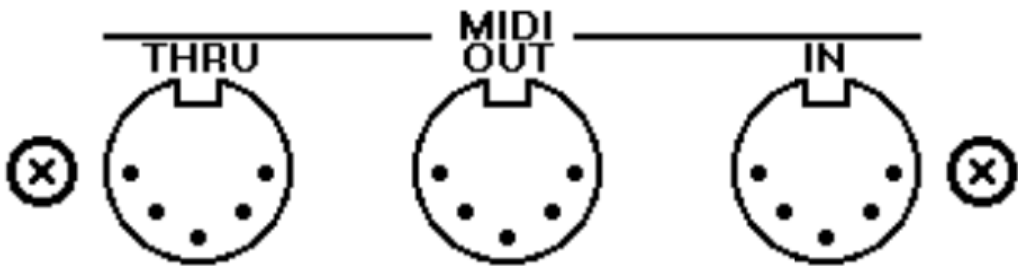
MIDI – Wie es funktioniert

- MIDI ist ein serielles Protokoll
 - Der Standard beschreibt sowohl die Hardware (MIDI-DIN) als auch die Software (MIDI Protocol)
 - Informationen werden Byte-weise übertragen
 - Ein oder mehrere Bytes ergeben das entsprechende Kommando
 - MIDI über Kabel: ein gepulstes Signal (wie Rechteck) mit 31.250 Hz
 - 1 MHz oder 500 kHz war gängiger CPU-Takt in den 80ern, $\frac{1}{32}$ oder $\frac{1}{16}$ des Taktes
 - Alle modernen Prozessoren arbeiten mit einem Vielfachen von 1 MHz
 - Ein MIDI-Byte besteht aus 10 bit, also 3.125 Byte pro Sekunde, kein Clock-Signal
 - Ein typisches Kommando besteht aus 2-3 Byte, damit 1000-1500 Events pro Sekunde
→ 0,6 ms pro Event

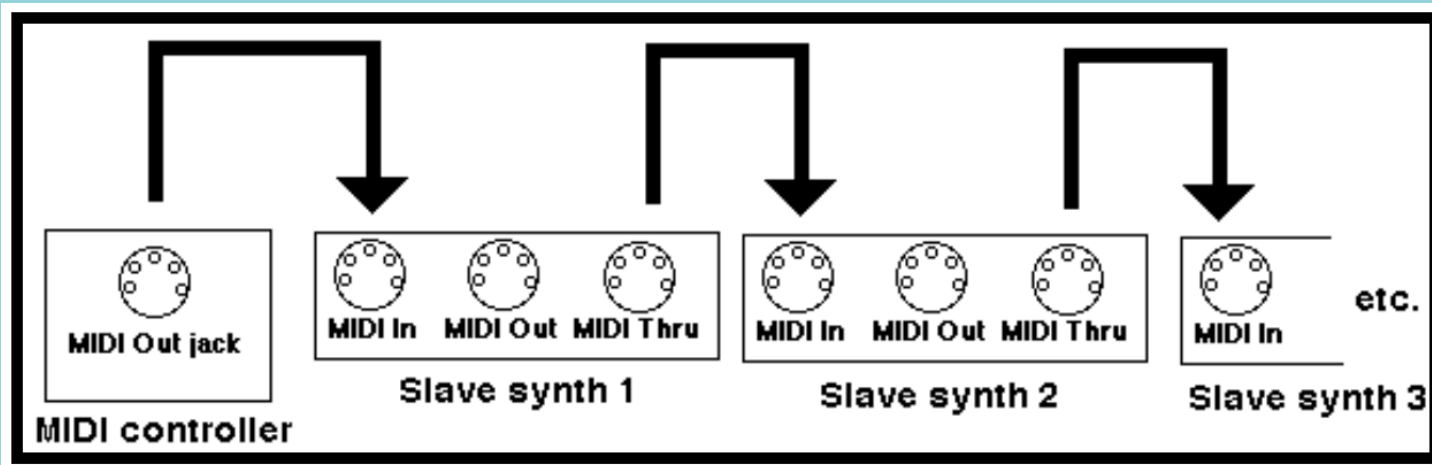
MIDI – Wie es funktioniert



MIDI – Wie es funktioniert



- Früher serielle, parallele Schnittstellen, Erweiterungsports.
- Heute USB → schneller, bidirektional, aber immer noch auf den Standard-MIDI Kommandos aufbauend
- USB nicht von Synth zu Synth



- Heute häufig MIDI ohne Kabel, alles in Software → Sequencer, Controller
- Virtuelle Instrumente
- Standards wie AU, VST, AAX usw.
- Alles läuft in MIDI V1.0 – nur ohne Kabel

MIDI – Wie es funktioniert → Kommando

- Am Keyboard: Eine Taste wird gedrückt
- In MIDI:
 - Byte 1 (Command or Status Byte): Note On
 - Byte 2 (Data Byte 1): Note number (which key)
 - Byte 3 (Data Byte 2): Velocity

Der MIDI Standard definiert jedes Status Byte sowie die Anzahl der darauf folgenden Datenbytes. Es gibt Kommandos mit 0 bis 2 Datenbytes. System exklusive Kommandos (SysEx) haben häufig eine größere Zahl von Datenbytes und sind teilweise komplex.

MIDI – Regeln

- HEX-Schreibweise ist nützlich
 - Erstes Bit == 1 → Command Byte → 80h oder größer
 - Erstes Bit == 0 → Data Byte → 7F oder kleiner
 - Wenn Command Byte mit 9 startet, dann „Note On“
 - Wenn Command Byte mit 8 startet, dann „Note Off“
- MIDI erlaubt 16 Kanäle
 - Geräte können Kanälen zugeordnet werden
 - Die unteren 4 Bit (rechte Stelle bei HEX Schreibweise) definieren den Kanal
 - Note On auf Kanal 8 → 97h

MIDI – Tabelle

<u>Name</u>	<u>Hex values</u> (channels 1-16)	<u>Decimal</u> <u>values</u>	<u>Data bytes</u>
Note Off	80-8F	128-143	2 (note number, velocity)
Note On	90-9F	144-159	2 (note number, velocity)
Key Pressure	A0-AF	160-175	2 (note number, pressure)
Control Change	B0-BF	176-191	2 (controller number, value)
Program Change	C0-CF	192-207	1 (program number)

Quelle: What is MIDI?, Paul D. Lehrman, 2017

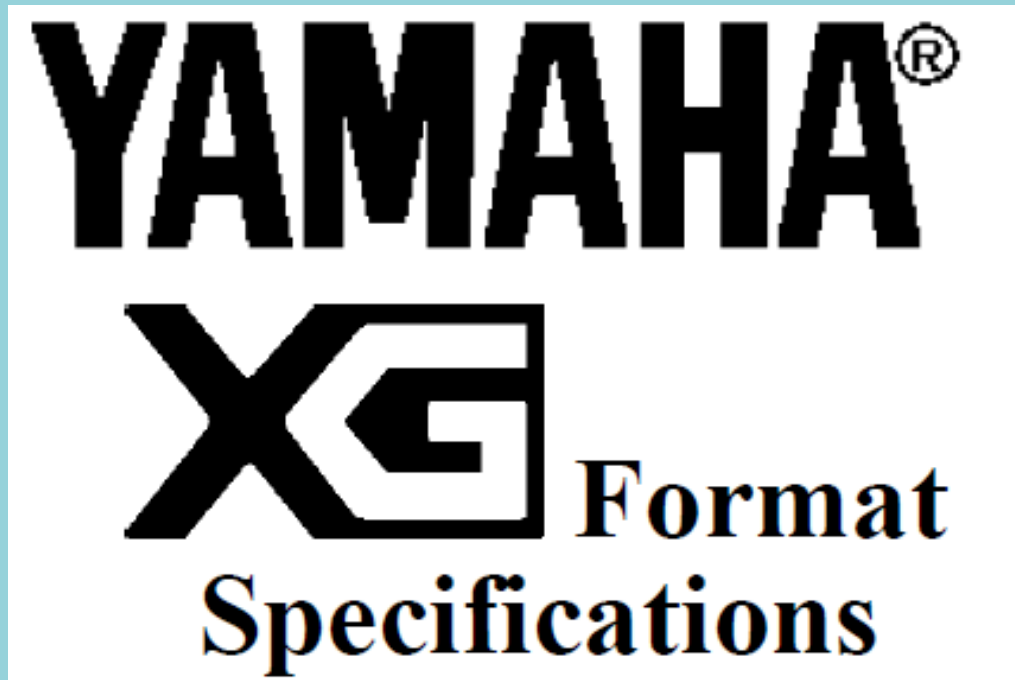
MIDI – Regeln

- Cxh → Program Change
 - anderes Instrument, Effektgeräte schalten von Hall auf Flanger usw.
 - Ein Data Byte, also 128 Instrumente
 - Moderne Geräte haben ein vielfaches an Instrumenten → Bank Select Kommando
- Bxh → Control Change
 - Modulation wheel, volume pedal
 - Controller 00 ist Bank Select
- Dxh → Channel Pressure
 - Axh → Polyphonic Key Pressure
- Exh → Pitch Bend
- F0h → Start System Exclusive – F7h Stop System Exclusive

Erweiterungen → General MIDI

- GM 1 (1991)
 - Gleiche Instrumente für jede Programmnummer
 - MIDI wurde herstellerübergreifend austauschbar
 - Mindestanzahl an Stimmen
 - Unterstützung von 16 Kanälen
 - Kanal 10 ist für Drum reserviert
- GM 2 (1999)
 - Erweiterungen für mehr Controllertypen, Effekte usw.

Erweiterungen: YAMAHA



Interessante Links

- [MIDI – History](#)
- [Home \(midi.org\)](#)

YAMAHA - Demo

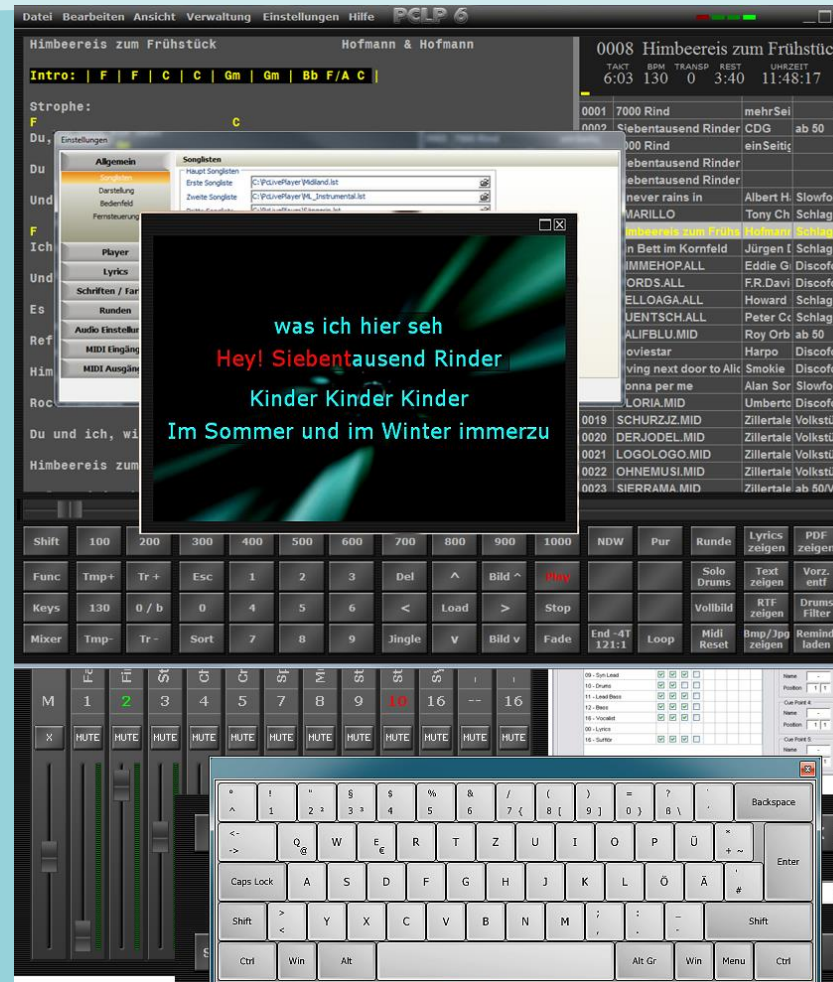
- MIDI erzeugen
 - Einzelschritt
 - Echtzeitaufnahme
- Analyse der Daten
- Blick in die Datenliste des Tyros 5

MIDI – Steuerung

Warum macht man es selbst?



Quelle: www.midiland.de



Quelle: <https://www.songbookplus.com>

YAMAHA – Steuerung per Tablet

```
<?xml version="1.0" standalone="yes"?>
<Threefour5>
  <xs:schema id="Threefour5" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="Threefour5" msdata:IsDataSet="true" msdata:Locale="en-US">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="Liste">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Titel" type="xs:string" default="" />
                <xs:element name="Interpret" type="xs:string" default="" />
                <xs:element name="Erscheinungsjahr" type="xs:string" default="" />
                <xs:element name="Jahrzent" type="xs:string" default="" />
                <xs:element name="Genre" type="xs:string" default="" />
                <xs:element name="Laufzeit" type="xs:string" default="" />
                <xs:element name="Gruppe" type="xs:string" default="" />
                <xs:element name="PDF" type="xs:string" default="" />
                <xs:element name="Registration" type="xs:string" default="" />
                <xs:element name="MIDI" type="xs:boolean" default="false" />
                <xs:element name="MP3" type="xs:boolean" default="false" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

YAMAHA – Steuerung per Tablet

```
<Liste>
  <Titel>Crazy Little Thing Called Love</Titel>
  <Interpret>Queen</Interpret>
  <Erscheinungsjahr>1979</Erscheinungsjahr>
  <Jahrzent>1970</Jahrzent>
  <Genre />
  <Laufzeit>2:42</Laufzeit>
  <Gruppe />
  <PDF>CrazyLittleThingCalledLove</PDF>
  <Registration>CrazyLittleThingCalledLove</Registration>
  <MIDI>true</MIDI>
  <MP3>false</MP3>
</Liste>
<Liste>
  <Titel>Sway</Titel>
  <Interpret>Dean Martin</Interpret>
  <Erscheinungsjahr />
  <Jahrzent>1950</Jahrzent>
  <Genre />
  <Laufzeit>2:30</Laufzeit>
  <Gruppe />
  <PDF>Sway</PDF>
  <Registration>Sway</Registration>
  <MIDI>false</MIDI>
  <MP3>false</MP3>
</Liste>
```


YAMAHA – Steuerung per Tablet

Bank Select MSB (CC 0)	Bank Select LSB (CC 32)	Registration Bank No. (at the beginning of the Reg file name)
0	0 ... 127	001 ... 128
1	0 ... 127	129 ... 256
2	0 ... 127	257 ... 384
3	0 ... 115	385 ... 500

Program Change 0 ... 7 call up Registration buttons 1 ... 8.

Quelle: <https://www.psrtutorial.com/forum/index.php/topic,46630.msg366662.html#msg366662>

YAMAHA – Steuerung per Tablet

```
if (midiPort >= 0)
{
    msb = System.Convert.ToByte(((posReg) / 128));
    lsb = System.Convert.ToByte(posReg % 128);
    midiOut.Open(midiPort);
    midConMess = midMessFac.CreateControllerMessage(channel, MidiControllerType.BankSelect, msb);
    midiOut.ShortData(midConMess.Data);
    midConMess = midMessFac.CreateControllerMessage(channel, MidiControllerType.BankSelectFine, lsb);
    midiOut.ShortData(midConMess.Data);
    midProgchange = midMessFac.CreateChannelMessage(MidiChannelCommand.ProgramChange, channel, 0, 0);
    midiOut.ShortData(midProgchange.Data);
    midiOut.Close();
}
```

YAMAHA – Steuerung per Tablet

```
midiOut.ShortData(midConMess.Data); ≤2ms elapsed
midConMess = midMess;
midiOut.ShortData(midConMess.Data);
midConMess = midiOut.ShortData(midConMess.Data);
midProgchar = midMessFac.CreateCommand(MidiChannelCom
midiOut.ShortData(midProgchange.Data); ≤2ms elapsed
midiOut.Close();
midiOut.Clos
```

midProgchange {CannedBytes.Midi.Mes

ByteLength	2
Command	ProgramChange
Data	192
MidiChannel	0
Parameter1	0
Parameter2	0
Status	192

```
ivate_Click(o
d ShowMenubarivate_Click(object, RoutedEventArgs)
id ShowMenubarid ShowMenubar_Click(object
dfViewer.pdfvi
fViewer.pdfviewer1.ShowTool
```

YAMAHA - Demo

- Abspielen in Media-Player
 - Einfaches GM
 - Kompliziertes T5
 - Abspielen auf T5

MIDI 2.0

Key Features of the MIDI 2.0 Environment

The Three Bs

There are three basic principles of future MIDI expansion based on MIDI Capability Inquiry



Bidirectional

MIDI-CI assumes
bidirectional communication
so MIDI changes from a
monologue to a dialogue



Backwards Compatible

MIDI-CI creates a way to separate
old devices and new devices and
always allows fallback to MIDI 1.0
maintaining compatibility



Both

One of the core goals of the MIDI
2.0 initiative is to also enhance the
MIDI 1.0 feature set whenever
possible.

Quelle: <https://www.midi.org/midi-articles/details-about-midi-2-0-midi-ci-profiles-and-property-exchange>

MIDI 2.0

Key Features of the MIDI 2.0 Environment

Three Ps

Future MIDI Expansion includes three main areas enabled by MIDI Capability Inquiry



Profile Configuration

Profiles are defined sets of rules for how a MIDI device sends or responds to specific MIDI messages to achieve a specific purpose or suit a specific application.



Property Exchange

Property Exchange (PE) messages can get and set device properties including but not limited to product name, configuration settings, controller names, controller values, patch names and other meta data, etc.



Protocol Negotiation

MIDI-CI defines how to negotiate to MIDI 2.0. Devices that do not support MIDI 2.0 will continue to use MIDI 1.0.

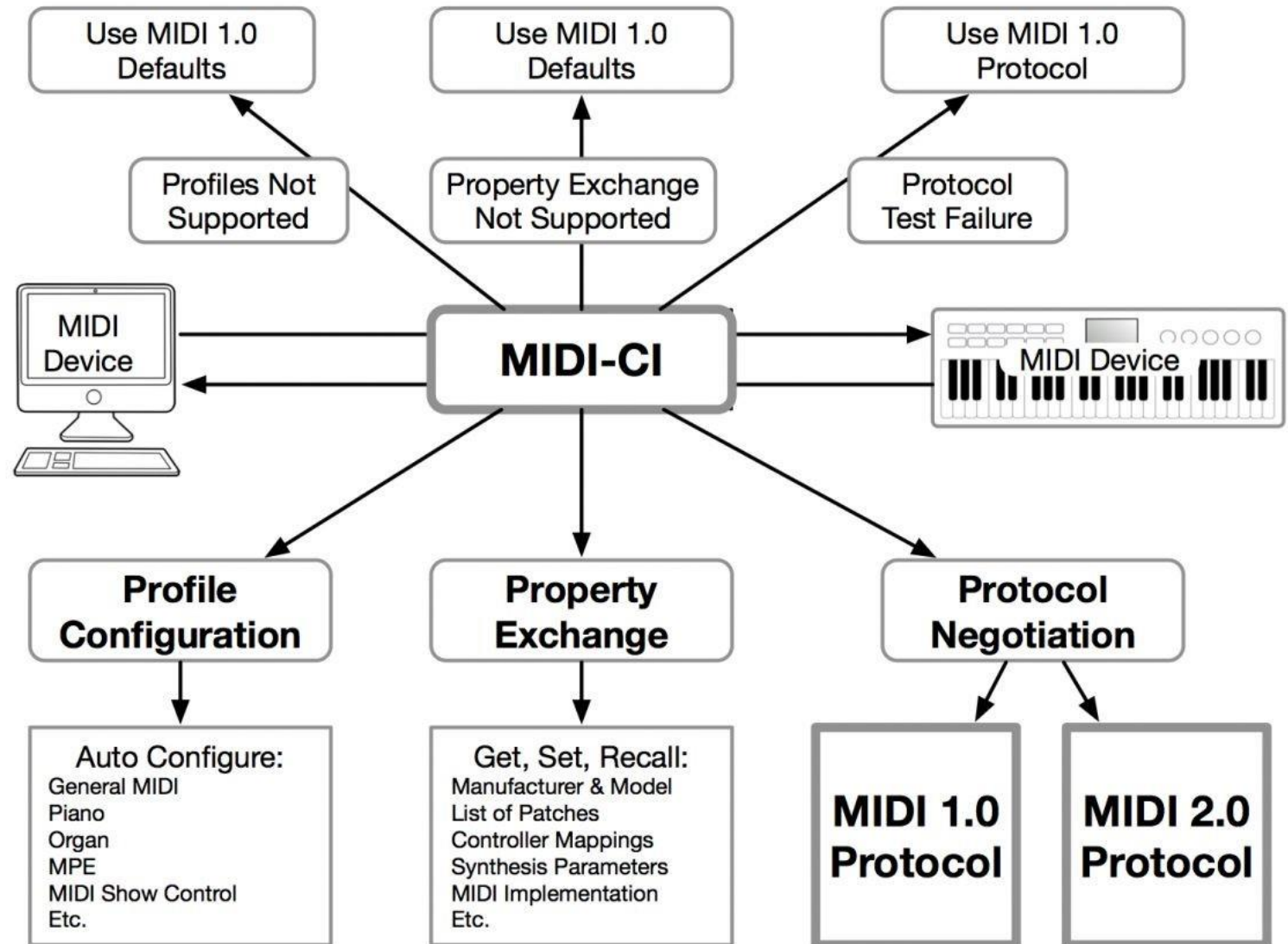
Not official logos

Quelle: <https://www.midi.org/midi-articles/details-about-midi-2-0-midi-ci-profiles-and-property-exchange>

MIDI 2.0

Quelle: <https://www.midi.org/midi-articles/details-about-midi-2-0-midi-ci-profiles-and-property-exchange>

MIDI 2.0 Environment



YAMAHA – Demo Tyros 5

Lust drauf?

MIDI

Vielen Dank für die Geduld! 😊