Erik Bartmann

# Modulare Synthesizer mit VCV-Rack 2 entdecken


## Was ist MIDI?

| | |
|---|---|
| Autor | Erik Bartmann |
| Internet | https://erik-bartmann.de/ |
| Thema | Grundlagen zum Thema MIDI |
| Version | 1.01 |
| Datum | 21. Januar 2022 |

# Inhaltsverzeichnis

# Basiswissen zu MIDI

## Worum geht es überhaupt?

In diesem Papier werden folgende Themen besprochen.

- Geschichtliches
- Was ist MIDI?
- MIDI IN, MIDI OUT, MIDI THRU
- Das MIDI-Protokoll
- Die MIDI-Kanäle
- Die MIDI-Informationen
- Das Programm MidiView

Elektronische Musikinstrumente wie Synthesizer sind schon eine feine Sache und es bereitet großen Spaß, sich damit auseinanderzusetzen. Baut man sich ein kleines Musikstudio auf, dann wird sich das sicherlich mit der Zeit ein wenig vergrößern, weil hier und da andere Geräte oder Instrumente hinzukommen. Hier ein Drum-Computer, da ein Mischpult oder auch Effektgeräte runden das Geschehen ab. Natürlich können derartige Instrumente sicherlich für sich alleine gesehen schon eine Menge Freude bereiten, doch erst im Zusammenspiel miteinander wird es dann richtig spannend. Es ist dann natürlich erforderlich, dass eine gewisse Kommunikation untereinander stattfindet, denn wenn zusätzlich auch noch Software in Form einer *DAW* (Digital Audio Workstation) mit von der Partie ist, dann sollte alles in irgendeiner Art und Weise synchronisiert werden. Da das Problem der Kommunikation verschiedener Geräte oder Instrumente untereinander schon sehr früh erkannt wurde, hat man sich Gedanken über eine Schnittstelle gemacht, über die Informationen ausgetauscht werden können. Werden also zum Beispiel verschiedene Tasten auf einem Synthesizer gedrückt, dann sollen ja auch entsprechend Töne erklingen. Wie sollte aber das ganze ablaufen, wenn zum Beispiel auf einem Masterkeyboard eine Taste gedrückt wird und in einem modularen System dann ein Klanggenerator in Form eines *VCO* (Voltage Controlled Oszillator) darauf zu reagieren hat? Die Abkürzung VCO beinhaltet schon die Lösung, denn es handelt sich um Spannungen, die dafür sorgen, dass bei den angeschlossenen Geräten etwas Entsprechendes ausgelöst wird. Diese sogenannte *CV* (Control-Voltage) wird also dafür verwendet, dass über analoge Spannungen zum Beispiel die Tonhöhe (Pitch) beeinflusst werden kann. Aber nicht nur die Tonhöhe, sondern auch andere Parameter eines Synthesizers wie Lautstärke, Filterverhalten oder Hüllkurven können darüber manipuliert werden. Es ist also offensichtlich, dass jegliche Kommunikation über unterschiedliche Spannungspegel erfolgt. Hinsichtlich modularer Systeme, ist das natürlich eine

gängige Praxis, wobei natürlich auf mögliche Verluste der Spannungspegel aufgrund unterschiedlicher Kabellängen oder äußere Einflüsse geachtet werden muss. Beim Bestreben, mehrere unterschiedliche Instrumente miteinander zu verbinden, muss eine andere Lösung her. Es haben sich verschiedene Hersteller zusammengeschlossen, um eine standardisierte Schnittstelle zu entwickeln, die einfach und auch preiswert zu realisieren sein sollte. Ein Informatiker mit dem Namen *Dave Smith* war maßgeblich an dieser Entwicklung beteiligt und erhielt sogar 2005 eine Auszeichnung in der *Hall of Fame* der *Mix Foundation TECnology* für die Spezifikation dieser Schnittstelle. Doch um welche Schnittstelle handelt es sich denn überhaupt? Es geht um die sogenannte *MIDI*-Schnittstelle!

# Was ist MIDI

MIDI ist die Abkürzung für ***M**usical **I**nstruments **D**igital **I**nterface* und ist eine Technologie, worüber elektronische Musikinstrumente untereinander kommunizieren können. Alle Geräte, die diese Sprache sprechen, können also miteinander reden. Diese Verbindung sind recht einfach aufzubauen, denn es gibt lediglich eine Buchse mit fünf Pins (nur 3 werden genutzt), die intern in drei Arten von Buchsen unterteilt sind.

- MIDI IN
- MIDI OUT
- MIDI THRU

In die *MIDI IN*-Buchse werden Information in das jeweilige Gerät geleitet und aus der *MIDI OUT*-Buchse kommen entsprechend Informationen heraus. Die *MIDI-THRU*-Buchse leitet das Signal vom *IN* einfach weiter, was dazu genutzt werden kann, um mehrere Geräte hintereinander zu schalten kann, die die gleichen Informationen erhalten sollen.

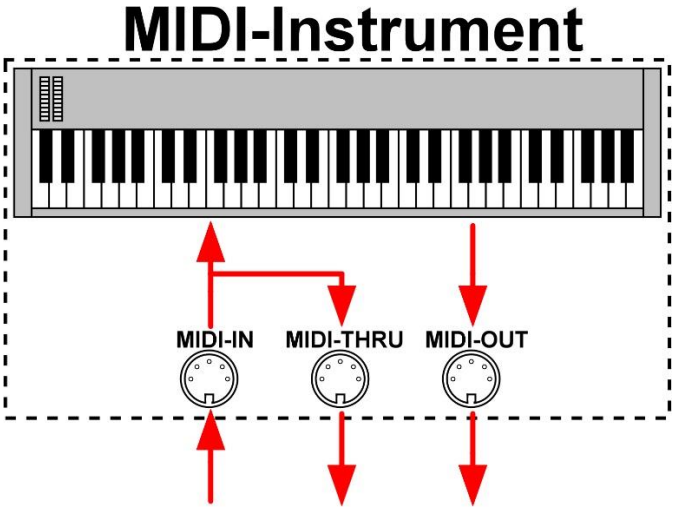


*Abbildung 1 Die drei MIDI-Buchsen*

Im Zeitalter von USB können MIDI-Informationen auch über die USB-Schnittstelle übertragen werden. Auf der folgenden Abbildung sind diese zwei Möglichkeiten untereinander zu sehen.

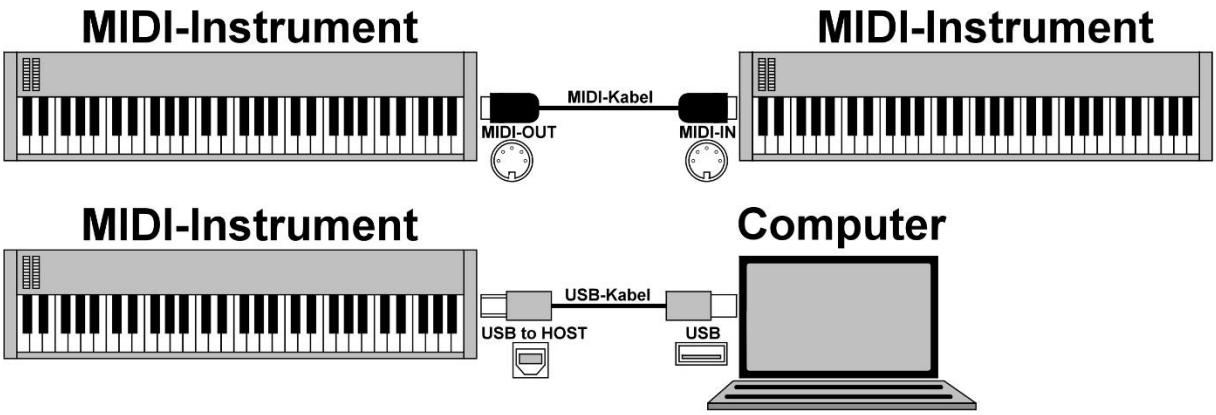


*Abbildung 2 Verschiedene MIDI-Verbindungen*

In welcher Weise kommunizieren aber die unterschiedlichen Geräte miteinander? Es handelt sich dabei um ein spezielles Protokoll, das extra für MIDI entwickelt wurde. Ausgangspunkt war die Situation, in der es darum ging, über ein Keyboard andere Geräte wie zum Beispiel Synthesizer anzusteuern. In den Anfängen der Entwicklung bestand die Lösung lediglich darin, mehr oder weniger lange Kabel zu verwenden, die analoge Signale übertrugen. Dieses Verfahren barg natürlich die schon angesprochenen Probleme hinsichtlich der Spannungsverluste in sich, die bei längeren Kabeln auftreten und zu sprichwörtlichen Unstimmigkeiten hinsichtlich von Tonhöhen führen konnten.

# Das MIDI-Protokoll

Die Kommunikation über die MIDI-Schnittstelle erfolgt in serieller Weise, also hintereinander mit einer Geschwindigkeit von 31250 Bit/s. Aber was wird da genau übertragen und in welcher Form? Ein MIDI-Befehl, der versendet wird, besteht aus drei Bytes.

### Was ist ein Byte beziehungsweise ein Bit?

Ein *Byte* ist eine Maßeinheit, die in der Digitaltechnik verwendet wird und eine Folge von 8 Bits bedeutet. Ein *Bit* ist dabei eine ein logischer Zustand, der entweder 0 oder 1 sein kann. 0 bedeutet, dass kein Strom fließt und 1, dass ein Strom fließt.

Diese drei Bytes beinhalten also alle relevanten Informationen, die ein angesteuertes MIDI-Gerät wie zum Beispiel ein Synthesizer braucht, damit er weiß, welches Modul er in welcher Note wie lange und wie laut spielen soll. Über MIDI werden keine Klänge versendet werden, wie zum Beispiel über eine Lautsprecherleitung, sondern lediglich Steuerbefehle. Das sollte dabei nicht vergessen werden! Wie schauen diese genannten drei Bytes aus und was für eine Bedeutung haben sie? Auf der folgenden Abbildung wird das verdeutlicht.

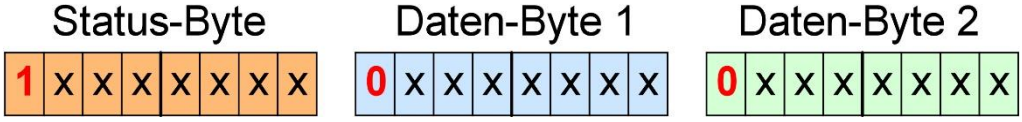


*Abbildung 3 Die drei MIDI-Bytes*

Bei der Versendung eines MIDI-Signals, das aus den drei genannten Bytes zusammengesetzt ist, handelt es sich um ein sogenanntes Event (Ereignis). Derartige Events können unterschiedlicher Natur sein. Sehen wir uns das genauer an.

# Ein MIDI-Ereignis

Tritt ein besonderes Ereignis auf, dann kann das unterschiedliche Ursachen haben. Es wird eine Taste auf dem Keyboard gedrückt und irgendwann wieder losgelassen oder es werden Drehregler bewegt. Diese Ereignisse besitzen bestimmte Ereignis-Zuweisungen, die in acht Kategorien beziehungsweise *Befehlstypen* unterteilt sind, wie das die folgende Tabelle bereitstellt.

| Befehlstyp | Status-Byte (binary) | Status (hex) |
|---|---|---|
| Note off | 1000 *nnnn* | 8*n* |
| Note on | 1001 *nnnn* | 9*n* |
| Poly Pressure | 1010 *nnnn* | A*n* |
| Control Change | 1011 *nnnn* | B*n* |
| Program Change | 1100 *nnnn* | C*n* |
| Channel Pressure | 1101 *nnnn* | D*n* |
| Pitch-Bend | 1110 *nnnn* | E*n* |

*Tabelle 1 MIDI-Befehlstypen*

Es ist zu sehen, dass sowohl eine binäre, als auch eine hexadezimale Schreibweise verwendet wird, was ich als Basiswissen an dieser Stelle jedoch voraussetze. Die gezeigten Informationen kommen beim ersten Byte, dem *Status-Byte* zum Tragen. Er werden der Befehlstyp und der MIDI-Kanal (1 bis 16) übermittelt, der durch *nnnn* gekennzeichnet ist. Das Status-Byte beginnt binär betrachtet immer mit einer 1. Das dient zur Unterscheidung zu den Daten-Bytes, die mit einer 0 beginnen. Wie schaut es mit den genannten MIDI-Kanälen aus? Die folgende Tabelle gibt Aufschluss.

| Binär | MIDI-Kanal | Binär | MIDI-Kanal |
|---|---|---|---|
| 0000 | 1 | 1000 | 9 |
| 0001 | 2 | 1001 | 10 |
| 0010 | 3 | 1010 | 11 |
| 0011 | 4 | 1011 | 12 |
| 0100 | 5 | 1100 | 13 |
| 0101 | 6 | 1101 | 14 |
| 0110 | 7 | 1110 | 15 |
| 0111 | 8 | 1111 | 16 |

*Tabelle 2 MIDI-Kanäle*

Es ist zu beachten, dass die binäre Zählweise bei 0 beginnt und dem MIDI-Kanal 1 zugewiesen ist. Sehen wir uns dazu ein Beispiel an.

## Note on

Angenommen, es wird das folgende MIDI-Ereignis hinsichtlich des Status-Bytes ausgelöst. Die beiden Daten-Bytes 1 und 2 spielen in diesem Fall keine Rolle.

***10010000***

Sehen wir uns das genauer an.

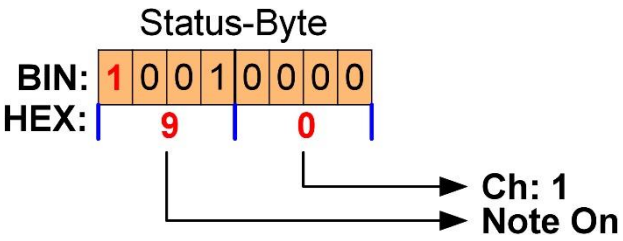


*Abbildung 4 Das Status-Byte für Note on*

Die linken 4 Bits zeigen, dass es sich um den HEX-Wert 9 handelt, der laut erster Tabelle der MIDI-Befehlstypen für das Ereignis *Note on* steht. Die rechte 4 Bits repräsentieren den gewünschten MIDI-Kanal, der hier mit dem HEX-Wert 0 zu sehen ist, was laut der zweiten Tabelle für die MIDI-Kanäle dem Kanal 1 entspricht. Wenn sich in einer *DAW* (Digital Audio Workstation) in einem MIDI-Clip eine Note befindet, dann handelt es sich im Grunde genommen um zwei getrennte MIDI-Befehle. *Note on* für das Drücken einer Taste und *Note off* für das Loslassen der Taste am Keyboard.

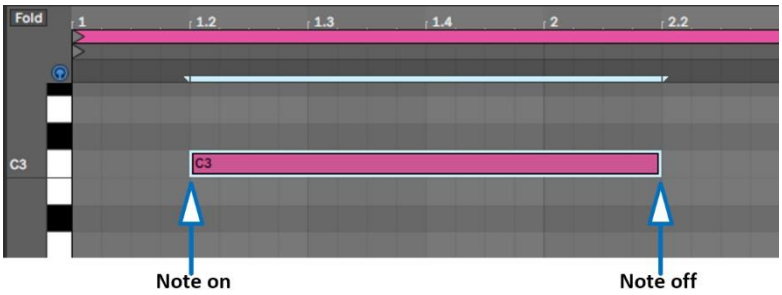


*Abbildung 5 Note on und Note off in einer DAW*

Die Länge der Note wird durch die Länge des grafischen Balkens abgebildet und befindet sich genau zwischen den beiden MIDI-Befehlen *Note on* und *Note off*. Nun sind aber noch weitere Informationen erforderlich, denn es muss hinsichtlich der Note noch die Notennummer und die Velocity (Anschlagstärke) übermittelt werden.

## Control-Change

Kommen wir nun zu einem weiteren Beispiel, bei dem es um die Steuerung eines Instrumentes hinsichtlich eines bestimmten Parameters geht. Das Ereignis wird Control-Change genannt und mit *CC* abgekürzt. Hier kommen die zwei Daten-Bytes ins Spiel, denn es müssen bei einem CC weitere Informationen wie *Controllernummer* und der entsprechende *Wert* für die Änderung übertragen werden. Angenommen, es wird an einem Keyboard oder einem Controller ein Drehregler bewegt, wie das auf der folgenden Abbildung zu sehen ist. Wie schaut das bei dem entsprechenden MIDI-Event aus?


*Abbildung 6 Drehregler am Keyboard*

Wie erwähnt, werden alle drei Bytes benötigt. Das folgende MIDI-Ereignis wird registriert.

***10110000 01000110 00100110***

Wie schaut das im Detail aus? Das erste von links ist wieder das Status-Byte, das sich wie folgt gestaltet.

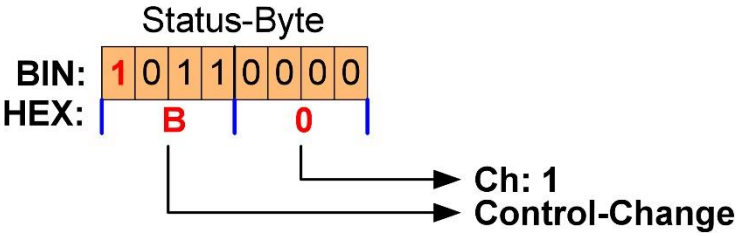


*Abbildung 7 Das Status-Byte für CC*

Die linken 4 Bits zeigen, dass es sich um den HEX-Wert B handelt, der laut erster Tabelle für MIDI-Befehlstypen für das Ereignis *Control-Change* steht. Die rechten 4 Bits repräsentieren den gewünschten MIDI-Kanal, der hier wieder mit dem HEX-Wert 0 zu sehen ist, was laut der zweiten Tabelle für die MIDI-Kanäle dem Kanal 1 entspricht. Zusammengesetzt ergibt das den hexadezimalen Wert B0, der in dezimaler Schreibweise 176 lautet. Man kann also hinsichtlich der einzelnen Kanäle folgendes festhalten.

- 176 + 0: Kanal 1
- 176 + 1: Kanal 2
- 176 + 2: Kanal 3
- usw.

Nun ist es wichtig zu wissen, welches Bedienelement (Controller) denn überhaupt bewegt wurde, was über das Daten-Byte 1 bestimmt wird. Durch die Einschränkung des Wertes auf 7 Bits - das linke und höchstwertige Bit besitzt immer den Wert 0 -, können Werte im Bereich zwischen 0 und 127 (128 Werte) erreicht werden.

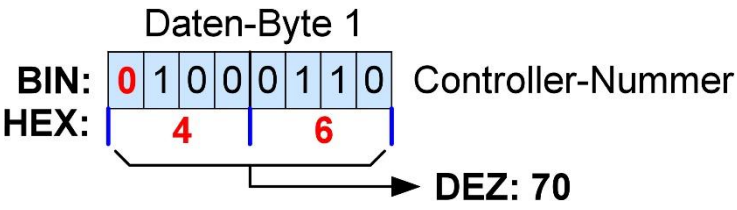


*Abbildung 8 Das Daten-Byte 1 für die Controller-Nummer*

Der hexadezimale Wert 46 entspricht dem dezimalen Wert 70. Und dem gerade von mir gedrehten Drehregler ist intern der dezimale Wert 70 als Controller-Nummer zugewiesen.

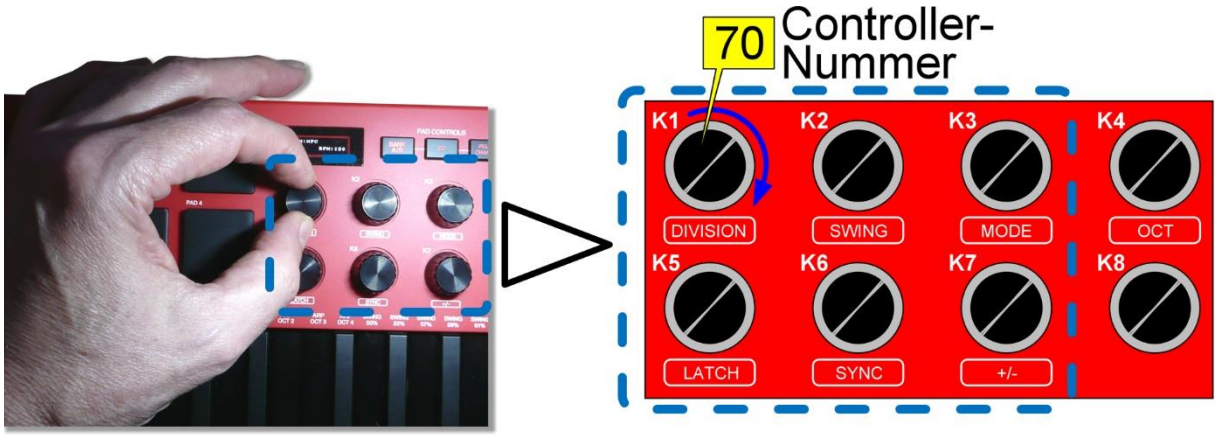


*Abbildung 9 Der Drehregler mit dem dezimalen Wert 70*

Da es einheitliche Bedienelemente an Synthesizern beziehungsweise Controllern gibt, sind bestimmte Werte reserviert, die nicht genutzt werden können. Ein Beispiel dazu wäre das sogenannte *MOD-Wheel* mit dem Wert 01. Nun kommt es noch darauf an, welche Stellung der Drehregler denn besitzt, sie über das Daten-Byte 2 abgebildet wird.

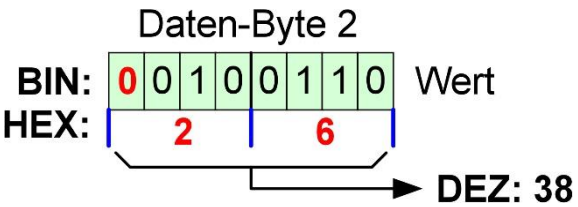


*Abbildung 10 Das Daten-Byte 2 für den Wert*

Hier ist es der hexadezimale Wert 26 der in dezimaler Schreibweise 38 ergibt. Auch in diesem Fall können durch die Einschränkung des Wertes auf 7 Bits nur Werte im Bereich zwischen 0 und 127 (128 Werte) ermittelt werden. Hinsichtlich des doch sehr geringen Wertebereiches von 128 unterschiedlichen Werten gibt es die Möglichkeit, zwei gleiche CC-Events hintereinander abzusetzen. Das verdoppelt den Wertebereich von 7-Bits (128-Werte) auf 14-Bits (16.384 Werte), was schon eine enorme Steigerung darstellt und eine viel weichere Abstufung ermöglicht.

# Ein MIDI-Monitor-Programm

Vielleicht ist es interessant zu sehen, wie derartige MIDI-Informationen von einem Keyboard oder Controller versendet werden und dazu gibt es ein freies Programm mit Namen *MIDIView*, das unter der folgenden Internetadresse zu bekommen ist.

**Hyperlink!**

https://hautetechnique.com/midi/midiview/

Ich habe also mein Keyboard in *MidiView* ausgewählt und den Drehregler bewegt, den ich eben schon verwendet habe.

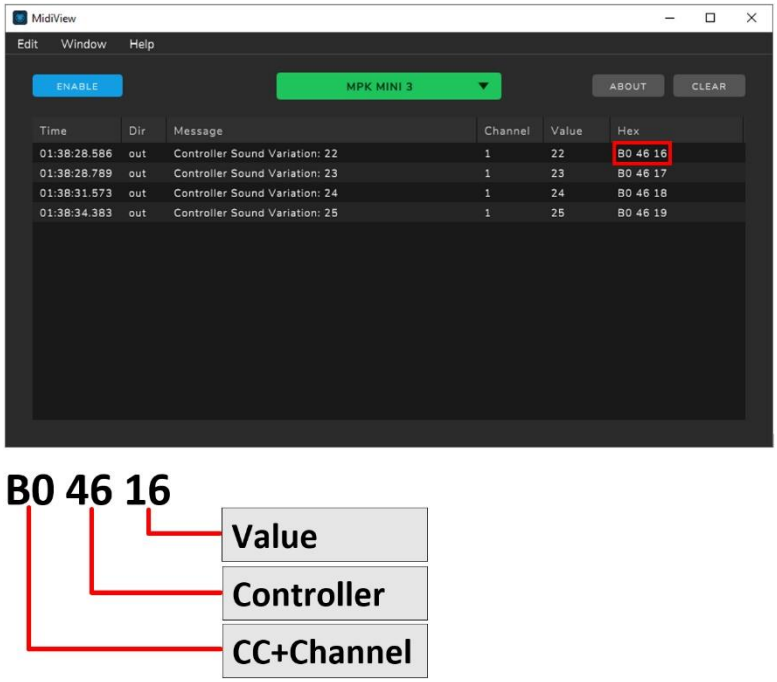


*Abbildung 11 Das Programm MidiView mit konkreten Werten*

Es ist sehr gut zu beobachten, welche Informationen dort auflaufen. Weitere Informationen sind auf meiner Internetseite zu finden.

**Hyperlinks!**

https://erik-bartmann.de/

https://erik-bartmann.de/?Musik___VCV-Rack

*Frohes Frickeln!*

Erik Bartmann